MODELING REAL-TIME HUMAN-AUTOMATION COLLABORATIVE SCHEDULING OF UNMANNED VEHICLES

by

ANDREW S. CLARE

S.B. Aerospace Engineering with Information Technology, Massachusetts Institute of Technology, 2008
S.M. Aeronautics and Astronautics, Massachusetts Institute of Technology, 2010

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author: _____
Andrew Clare, Dept. of Aeronautics and Astronautics
May 23, 2013

Certified by: _____
Mary L. Cummings, Thesis Supervisor
Associate Professor of Aeronautics and Astronautics
Director, Humans and Automation Laboratory

Certified by: _____
Emilio Frazzoli
Associate Professor of Aeronautics and Astronautics

Certified by: _____
Nelson P. Repenning
Professor of Management Science and Organizations Studies

Accepted by: _____
Eytan H. Modiano
Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

1. REPORT DATE **JUN 2013**	2. REPORT TYPE	3. DATES COVERED **00-00-2013 to 00-00-2013**
4. TITLE AND SUBTITLE **Modeling Real-Time Human-Automation Collaborative Scheduling of Unmanned Vehicles**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Massachusetts Institute of Technology,Department of Aeronautics and Astronautics,Cambridge,MA,02139**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		

14. ABSTRACT

Recent advances in autonomy have enabled a future vision of single operator control of multiple heterogeneous Unmanned Vehicles (UVs). Real-time scheduling for multiple UVs in uncertain environments will require the computational ability of optimization algorithms combined with the judgment and adaptability of human supervisors. Automated Schedulers (AS), while faster and more accurate than humans at complex computation, are notoriously ?brittle? in that they can only take into account those quantifiable variables, parameters, objectives, and constraints identified in the design stages that were deemed to be critical. Previous research has shown that when human operators collaborate with AS in real-time operations, inappropriate levels of operator trust, high operator workload, and a lack of goal alignment between the operator and AS can cause lower system performance and costly or deadly errors. Currently, designers trying to address these issues test different system components, training methods, and interaction modalities through costly human-in-the-loop testing. Thus, the objective of this thesis was to develop and validate a computational model of real-time human-automation collaborative scheduling of multiple UVs. First, attributes that are important to consider when modeling real-time human-automation collaborative scheduling were identified, providing a theoretical basis for the model proposed in this thesis. Second, a Collaborative Human-Automation Scheduling (CHAS) model was developed using system dynamics modeling techniques, enabling the model to capture non-linear human behavior and performance patterns, latencies and feedback interactions in the system, and qualitative variables such as human trust in automation. The CHAS model can aid a designer of future UV systems by simulating the impact of changes in system design and operator training on human and system performance. This can reduce the need for time-consuming human-in-the-loop testing that is typically required to evaluate such changes. It can also allow the designer to explore a wider trade space of system changes than is possible through prototyping or experimentation. Through a multi-stage validation process, the CHAS model was tested on three experimental data sets to build confidence in the accuracy and robustness of the model under different conditions. Next, the CHAS model was used to develop recommendations for system design and training changes to improve system performance. These changes were implemented and through an additional set of human subject experiments, the quantitative predictions of the CHAS model were validated. Specifically, test subjects who play computer and video games frequently were found to have a higher propensity to over-trust

15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**336**	

MODELING REAL-TIME HUMAN-AUTOMATION COLLABORATIVE SCHEDULING OF UNMANNED VEHICLES

by

Andrew S. Clare

Submitted to the Department of Aeronautics and Astronautics on May 23rd, 2013 in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Aeronautics and Astronautics.

Abstract

Recent advances in autonomy have enabled a future vision of single operator control of multiple heterogeneous Unmanned Vehicles (UVs). Real-time scheduling for multiple UVs in uncertain environments will require the computational ability of optimization algorithms combined with the judgment and adaptability of human supervisors. Automated Schedulers (AS), while faster and more accurate than humans at complex computation, are notoriously "brittle" in that they can only take into account those quantifiable variables, parameters, objectives, and constraints identified in the design stages that were deemed to be critical. Previous research has shown that when human operators collaborate with AS in real-time operations, inappropriate levels of operator trust, high operator workload, and a lack of goal alignment between the operator and AS can cause lower system performance and costly or deadly errors. Currently, designers trying to address these issues test different system components, training methods, and interaction modalities through costly human-in-the-loop testing.

Thus, the objective of this thesis was to develop and validate a computational model of real-time human-automation collaborative scheduling of multiple UVs. First, attributes that are important to consider when modeling real-time human-automation collaborative scheduling were identified, providing a theoretical basis for the model proposed in this thesis. Second, a Collaborative Human-Automation Scheduling (CHAS) model was developed using system dynamics modeling techniques, enabling the model to capture non-linear human behavior and performance patterns, latencies and feedback interactions in the system, and qualitative variables such as human trust in automation. The CHAS model can aid a designer of future UV systems by simulating the impact of changes in system design and operator training on human and system performance. This can reduce the need for time-consuming human-in-the-loop testing that is typically required to evaluate such changes. It can also allow the designer to explore a wider trade space of system changes than is possible through prototyping or experimentation.

Through a multi-stage validation process, the CHAS model was tested on three experimental data sets to build confidence in the accuracy and robustness of the model under different conditions. Next, the CHAS model was used to develop recommendations for system design and training changes to improve system performance. These changes were implemented and through an additional set of human subject experiments, the quantitative predictions of the CHAS model were validated. Specifically, test subjects who play computer and video games frequently were found to have a higher propensity to over-trust automation. By priming these gamers to lower their initial trust to a more appropriate level, system performance was improved by 10% as compared to gamers who were primed to have higher trust in the AS. The CHAS model provided accurate quantitative predictions of the impact of priming operator trust on system performance. Finally, the boundary conditions, limitations, and generalizability of the CHAS model for use with other real-time human-automation collaborative scheduling systems were evaluated.

Thesis Supervisor: Mary L. Cummings
Title: Associate Professor of Aeronautics and Astronautics

Acknowledgments

I owe thanks to numerous people for the successful completion of this thesis.

First, to my advisor Missy Cummings, for always believing in me over our long journey together. You took me on as an undergraduate researcher over 6 years ago. You encouraged me to go to graduate school and apply for my fellowship. You recognized my potential early on, constantly pushing me to grow and develop in ways that are remarkable to me now as I look back. For all of the opportunities that you provided, all of the advice, both personal and professional, that you gave, and for all of the paper, proposal, and thesis editing that you suffered through, I will always be grateful.

To my committee members, Nelson Repenning and Emilio Frazzoli. Nelson, thank you for taking a risk by joining my interdisciplinary thesis committee and venturing to our side of the campus. You provided more help and guidance in understanding the intricacies of System Dynamics modeling than I ever could have hoped for. Emilio, thank you for advising me throughout this process to always take a step back and keep the bigger picture in mind with my thesis work.

To my thesis readers, Mike Lewis and John Carroll, for volunteering your valuable time to provide such thoughtful comments and suggestions.

To Julie Shah, for your caring advice and encouragement throughout my graduate school career. Thank you for seamlessly taking me under your wing for the past two years.

To Wesley Harris, for sharing your sage advice and wisdom with me since before I even enrolled at MIT, when I first met you as a visiting high school student. Thank you for all of the doors that you've opened and for all of the opportunities you pushed me to pursue. You've served as my academic advisor and my housemaster and I hope that you'll continue to serve as a valued mentor into the future.

To everyone who contributed to the development of the OPS-USERS system: Olivier, Karl, Andrew K., Andrew W., Cameron, Christin, Dan, Scott, Ian, Pierre, and Armen. I learned so much from all of you, had a great time (even when things broke), and I couldn't have asked for better teammates on this project. Thank you also to Aurora Flight Sciences and the Aerospace Controls Laboratory for supporting OPS-USERS.

To Sally Chapman, Marie Stuppard, and Beth Marois for all of the advice, support, and help that you've provided through all of my years in AeroAstro.

To my fellow HALiens past and present: Jason, Armen, Christin, Dan, Kim, Paul, Dave, Geoff, Anna, Birsen, Luca, Yves, Sylvain, Ryan, Jason, Kathleen, Mariela, Scott, Luisa, Maarten, Ian, and Pierre and many others, thank you for the constant support, as well as all of the good times in and out of the lab.

To my Mom, Dad, and brother for your unwavering love and support through my (too many) years in school. You were the first to believe in me and your encouragement and advice have always lifted my spirits and pushed me to pursue my dreams.

Finally, to Brianne. You stuck with me through all of the ups and downs of MIT. You always knew when I was getting off track or when I was having a rough time. You never stopped believing, even when I wanted to. You always knew exactly what to do, whether it was encouragement, distraction, motivation, or setting a deadline. I can never thank you enough for all of your love and support. I can't wait to start our life together.

Table of Contents

List of Figures

14

15

List of Tables

List of Acronyms

ACT-R	Adaptive Control of Thought-Rational
ANOVA	Analysis of Variance
AS	Automated Scheduler
ATC	Air Traffic Control
CBBA	Consensus Based Bundle Algorithm
CHAS	Collaborative Human-Automation Scheduling
DARPA	Defense Advanced Research Projects Agency
DES	Discrete Event Simulation
DoD	Department of Defense
EP	Expected Performance
FAA	Federal Aviation Administration
GOMS	Goals, Operations, Methods, and Selection
MAI	Metacognitive Awareness Inventory
MANOVA	Multivariate Analysis of Variance
MAPE	Mean Absolute Percent Error
MIT	Massachusetts Institute of Technology
MSE	Mean Square Error
NAS	National Airspace System
NASA	National Aeronautics and Space Administration
NST	Nonscheduling Task Load
OPS-USERS	Onboard Planning System for UxVs Supporting Expeditionary Reconnaissance and Surveillance
PPG	Perceived Performance Gap
PPP	Perceived Present Performance
R^2	Coefficient of Determination
RMSE	Root Mean Square Error
ROE	Rules of Engagement
SA	Situation Awareness
SCT	Schedule Comparison Tool
SD	System Dynamics

SE	Standard Error
THEP	Time Horizon for Expected Performance
TPPP	Time to Perceive Present Performance
UAV	Unmanned Aerial Vehicle
U^C	Covariation component of MSE
U^M	Bias component of MSE
U^S	Variation component of MSE
USAR	Urban Search and Rescue
USV	Unmanned Surface Vehicle
UV	Unmanned Vehicle
UxV	Unmanned Vehicle of any type (air, land, sea)
WUAV	Weaponized Unmanned Aerial Vehicle

1 Introduction

Real-time scheduling in uncertain environments is crucial to a number of domains, including Air Traffic Control (ATC) (Wickens et al., 1998), rail operations (Kwon, Martland, & Sussman, 1998), manufacturing plants (Jackson, Wilson, & MacCarthy, 2004), space satellite control (Howe et al., 2000), and Unmanned Vehicle (UV) operations (Clare, 2010). The representative setting for this thesis will be UV operations, as the use of UVs has increased dramatically over the past decade (Girard & Hedrick, 2004; Naval Studies Board, 2005; U.S. Air Force, 2009). The United States Department of Defense (DoD) allocated $1.82 billion for UV research and development in its 2010 budget, with general UV funding increasing 18% from 2009 (Washington Technology, 2009). Beyond military operations, UVs were used in natural disaster relief efforts and in the response to the Fukushima nuclear disaster (Madrigal, 2011; Thomas, 2012). Recently, the U.S. Congress passed legislation (H.R. 658, 2012) mandating the integration of commercial Unmanned Aerial Vehicles (UAVs) into the U.S. National Airspace System (NAS) by 2015. UAV integration into the NAS is projected to have an $82 billion economic impact over 10 years (Jenkins, 2013), with uses for UAVs including agriculture, wildlife monitoring, firefighting, search and rescue, border patrol, atmospheric research, and cargo delivery.

While these UVs contain advanced technology, they typically require multiple human operators, often more than a comparable manned vehicle would require (Haddal & Gertler, 2010). The need for many operators per UV causes increased training and operating costs (Haddal & Gertler, 2010) and challenges in meeting the ever-increasing demand for more UV operations (U.S. Air Force, 2009). For nearly a decade, the U.S. DoD has envisioned inverting the operator-to-vehicle ratio in future scenarios where a single operator controls multiple heterogeneous (air, sea, land) UVs simultaneously (Naval Studies Board, 2005; Office of the Secretary of Defense, 2005). More recently, the DoD made it clear that increasing UV autonomy and developing advanced techniques for manned-unmanned teaming are priorities for future research in the "Unmanned Systems Integrated Roadmap FY2011-2036" (DoD, 2011). The roadmap discussed the need for more advanced autonomy, the desire for this autonomy to be flexible and adaptable to dynamic and uncertain environments, the need for collaborative autonomy between multiple UVs, and the need for new verification and validation approaches to certify increasingly complex autonomy.

Recent advances in the autonomous capabilities of UVs have enabled these vehicles to execute basic operational and navigational tasks on their own and collaborate with other UVs to complete higher level tasks, such as surveying a designated area (Alighanbari & How, 2006; Bertuccelli et al., 2009). Researchers have demonstrated autonomous collaboration and swarming behavior with anywhere from three to hundreds of vehicles (see Mohan & Ponnambalam, 2009 for a review). In October 2012, NASA and DARPA demonstrated advanced collaboration between multiple UVs for autonomous aerial refueling between two unmanned, high-altitude Global Hawk aircraft, as shown in Figure 1 (NASA, 2012).

Figure 1. Demonstration of autonomous multi-UV collaboration for aerial refueling (NASA, 2012).

To effectively control multiple semi-autonomous UVs, some method is necessary for scheduling tasks. For the purposes of this thesis, scheduling is defined as creating a temporal plan that assigns tasks among a team of UVs and determines when the tasks will be completed. While this thesis will not focus on path planning per se, it should be noted that path planning is coupled with the scheduling problem, due to the need to estimate how long it will take for a UV to travel to a certain location to accomplish a task. A wide variety of optimization algorithms have been developed to address the problem of scheduling tasks for multiple UVs (see Clare, Cummings, & Bertuccelli, 2012 for a review). While varying in their method of formulating the scheduling

28

problem and solving the optimization, all of the approaches cited in the above paper utilize an Automated Scheduler (AS) with little to no human input during the development of the schedule.

However, in the presence of unknown variables, possibly inaccurate information, and changing environments, automated scheduling algorithms do not always perform well (Scott, Lesh, & Klau, 2002; Silverman, 1992). Though fast and able to handle complex computation far better than humans, optimization algorithms are notoriously "brittle" in that they can only take into account those quantifiable variables, parameters, objectives, and constraints identified in the design stages that were deemed to be critical (Smith, McCoy, & Layton, 1997). In a command and control situation such as supervising multiple UVs, where unanticipated events such as weather changes, vehicle failures, unexpected target movements, and new mission objectives often occur, AS have difficulty accounting for and responding to unforeseen changes in the environment (Guerlain, 1995; Polson & Smith, 1999). Additionally, the designers of optimization algorithms often make a variety of assumptions when formulating the optimization problem, determining what information to take into account, or, in the case of receding horizon algorithms, deciding how far into the future to plan (Bellingham, Richards, & How, 2002; Layton, Smith, & McCoy, 1994). While the DoD is enthusiastic about autonomy, the 2011 roadmap also cautioned that:

> "Because artificial systems lack the human ability to step outside a problem and independently reevaluate a novel situation based on commander's intent, algorithms that are extremely proficient at finding optimal solutions for specific problems may fail, and fail badly, when faced with situations other than the ones for which they were programmed." (DoD, 2011, p. 48)

One approach to deal with the "brittleness" of these algorithms is to have a human operator and an algorithm work together. A 2012 Defense Science Board report, "The Role of Autonomy in DoD Systems," explicitly called for human-automation collaboration, stating:

> "It should be made clear that all autonomous systems are supervised by human operators at some level, and autonomous systems' software embodies the designed limits on the actions and decisions delegated to the computer. Instead of viewing autonomy as an intrinsic property of an unmanned vehicle in isolation, the design and operation of autonomous systems needs to be considered in terms of human-system collaboration." (U.S. Department of Defense, 2012, pp. 1-2)

A mixed-initiative scheduling system, where a human guides a computer algorithm in a collaborative process to solve the scheduling problem, could best handle a realistic scenario with unknown variables, possibly inaccurate information, and dynamic environments. A number of studies have shown that humans collaborating with algorithms can achieve higher performance than either the human or the algorithm alone under certain conditions (Anderson et al., 2000; Cummings et al., 2012; Cummings & Thornburg, 2011; Johnson et al., 2002; Malasky et al., 2005; Ponda et al., 2011; Ryan, 2011). Designing a more effective human-automation collaborative scheduling system would provide the ability to supervise multiple UVs while addressing the inherent brittleness and opacity of algorithms.

1.1 Motivation

Despite the potential benefits of a collaborative scheduling system, operators can become confused when working with automation, unaware of how the "black box" algorithm came to its solution or the assumptions made by the algorithm in modeling the problem. Often times there are differences between the real world, the automation/engineer's model, and the human operator's models of the world (Figure 2).
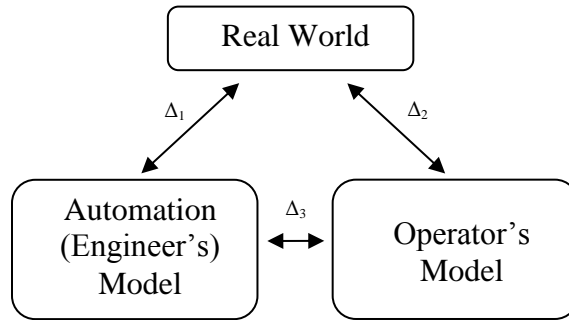
Figure 2. Differences between real world, the algorithm/engineer's model, and operator's model of the world.

Three major problems have been identified when human operators collaborate with AS in real-time operations due to the "brittleness" of the AS. First, operator trust in the AS can fluctuate due both to the operator's initial trust level in the AS and the behavior of the AS throughout the mission. This phenomenon has been observed in data analysis from previous human-in-the-loop experiments and has been linked to changes in performance (Clare, Macbeth, & Cummings, 2012; Gao et al., 2013). Trust can be defined as the "attitude that an agent will help achieve an

individual's goals in a situation characterized by uncertainty and vulnerability" (Lee & See, 2004, p. 51). Operators with low trust may spend an excessive amount of time replanning or adjusting the schedule (Clare, Macbeth, et al., 2012). Also, "overtrust" in automation has been cited in a number of costly and deadly accidents in a variety of domains (Cummings, 2004a; Parasuraman & Riley, 1997).

The second problem is that operator cognitive workload can reach a level where system performance begins to decline, as shown in previous research (Clare & Cummings, 2011; Cummings, Clare, & Hart, 2010). When the operator's workload becomes too high, he or she may not have enough spare attentional resources to evaluate the schedules generated by the AS and determine whether the schedules need to be adjusted. On the other hand, increased automation can lower the operator's task load to the point where boredom and fatigue can negatively impact performance (Cummings et al., 2013; Mkrtchyan, 2011; Scerbo, 2001; Walters, French, & Barnes, 2000).

The third problem is that there may be a lack of goal alignment between the operator and AS. Thus, the schedules generated by the AS may not be achieving the goals that are most important at that moment during the mission. There are two primary methods by which this could occur. First, the objective functions of the human and AS may not be aligned, in that the AS is optimizing for one objective, while the operator has a dynamic or different and possibly subjective metric for evaluating schedules. Previous research has shown that alignment of operator and AS goals can lead to higher Situation Awareness (SA), higher levels of operator confidence, and higher spare mental capacity (Clare et al., 2012). Second, the operator may be most concerned with satisficing (Simon et al., 1986), or achieving a feasible solution as quickly as possible, as opposed to finding the optimal schedule through lengthy optimization.

Understanding how to design collaborative scheduling systems to achieve the appropriate level of human trust, a moderate level of workload, and alignment of operator and AS goals is crucial to maintaining high system performance and preventing mistakes. Also, testing the impact of different system components (algorithms, interfaces, etc.) and interaction modalities on human and system performance typically requires costly and time-consuming human-in-the-loop testing. Thus, the objective of this thesis was to develop and validate a computational model of real-time

human-automation collaborative scheduling of multiple UVs that could be used to predict the impact of changes in system design and operator training on human and system performance.

1.2 Research Questions

To address this objective, the following research questions were posed:

- What are the major attributes and interactions that a model of real-time human-automation collaborative scheduling must capture?
- Can the model be used to predict the impact of changes in system design and operator training on human and system performance?
- What level of accuracy can be expected of this model? How does it compare to other relevant models? What are the boundary conditions of the model?

1.3 Thesis Organization

In order to address these research questions, this thesis has been organized into the following chapters:

- Chapter 1, *Introduction*, describes the motivation, objectives, and research questions of this thesis.
- Chapter 2, *Background*, defines the concept of real-time human-automation collaborative scheduling of multiple UVs. Previously developed relevant models of humans in scheduling and control situations are presented and three crucial gaps among these previous models are identified. Finally, six attributes that are important to consider when modeling real-time human-automation collaborative scheduling are elicited from the relevant body of literature.
- Chapter 3, *Model Development*, describes the modeling process that created the Collaborative Human-Automation Scheduling (CHAS) model, a System Dynamics (SD) model of human-automation collaborative scheduling of multiple UVs. The chapter describes the model in detail and concludes by describing the outputs and benefits of the model
- Chapter 4, *Model Validation*, presents the results of the multi-stage validation process that was conducted for the CHAS model. Model structure tests are described, including boundary adequacy testing, dimensional consistency testing, extreme conditions testing, integration error testing, and structure and parameter verification. Behavior reproduction testing to

evaluate the accuracy of the CHAS model is described for three data sets. A sensitivity analysis is presented to evaluate the robustness of CHAS model results.

- Chapter 5, *Predictive Validation Experiment*, describes a human subject experiment that was conducted to evaluate the ability of the CHAS model to predict the impact of changes in system design and operator training on human and system performance. The experimental results are presented, including an analysis of the impact of demographics on performance. Data is presented to evaluate the assumptions built into the CHAS model. Finally, the experiment results are compared to the quantitative predictions made by the CHAS model.

- Chapter 6, *Model Synthesis*, demonstrates potential uses for the CHAS model by system designers. The CHAS model's accuracy and features are compared with a previously developed Discrete Event Simulation (DES) model of human supervisory control of multiple UVs. Finally, the generalizability of the model is discussed along with model limitations.

- Chapter 7, *Conclusions*, summarizes the important results in the CHAS model's development and validation. Also, this chapter evaluates how well the research objectives were met, suggests potential future work, and presents the key contributions of this thesis.

2 Background: Modeling Human-Automation Collaborative Scheduling

This chapter begins by defining the concept of real-time human-automation collaborative scheduling of multiple Unmanned Vehicles (UVs). The unique aspects of this method of controlling UVs are identified along with the potential benefits to both human and system performance. Previously developed relevant models of humans in scheduling and control situations are then presented. Three crucial gaps among these previous models are identified with regards to real-time human-automation collaborative scheduling of multiple UVs. Finally, attributes that are important to consider when modeling real-time human-automation collaborative scheduling are elicited from the body of literature relevant to human supervisory control. These attributes provide a theoretical basis for the model proposed in this thesis.

2.1 Real-time Human-Automation Collaborative Scheduling

A potential future method for a single human operator to control multiple heterogeneous UVs (air, land, sea) involves the operator guiding an Automated Scheduler (AS) in a collaborative process to create, modify, and approve schedules for the team of UVs, which are then carried out by the semi-autonomous UVs. Although this concept is known by many names, including "Human-Automation Collaboration" (Miller & Parasuraman, 2003), "Human-Computer Collaboration" (Silverman, 1992), "Human Guided Algorithms" (Klau et al., 2003; Thorner, 2007), and "Mixed-initiative Planning" (Riley, 1989), all such systems involve a human working collaboratively with an optimization algorithm to solve a complex problem or make a decision. It is likely that operators will need to concentrate attention on the primary task of monitoring UV progress and system performance while also being prepared for various alerts, such automation notifications about potential changes to the vehicle schedules.

For the purposes of this thesis, the representative setting will be a reconnaissance mission to search for an unknown number of mobile targets, each of which may be friendly, hostile, or unknown. The mission scenario is multi-objective, and includes finding as many targets as possible, tracking already-found targets, and neutralizing all hostile targets. Scheduling is defined here as creating a temporal plan that assigns tasks/targets among the team of UVs and determines when the tasks will be completed. While this thesis will not focus on path planning

per se, it should be noted that path planning is coupled with the scheduling problem, due to the need to estimate how long it will take for a UV to travel to a certain location to accomplish a task.

This thesis will also focus on "real-time" scheduling. As opposed to the formal definition of real-time in the computer science field, which is a system that must guarantee a result within a hard time constraint (Stankovic, 1988), our definition of real-time involves mission planning and replanning on-the-fly due to a dynamic and uncertain environment. For example, tasks may appear or disappear, move, or obstacles may appear or change throughout the mission. This implies a soft constraint on the speed of the collaborative scheduling process; the human operator and the AS must be capable of creating and approving new plans rapidly enough to be able to replan while a mission is already underway, in contrast to "off-line" pre-planning of a mission. While this soft time constraint would be different for each possible application scenario, this thesis focuses on a highly dynamic environment where the AS must be capable of generating new plans within seconds.

This type of scheduling problem, assigning multiple heterogeneous UVs to many possible tasks with capability, location, and timing constraints in uncertain, dynamic environments, is likely a situation with unbounded indeterminacy (Russell & Norvig, 2003), depending on the problem formulation, where the set of possible preconditions or effects either is unknown or is too large to be enumerated completely. This optimization problem is NP-hard, meaning that it is likely that the AS cannot find an optimal solution in polynomial time (Russell & Norvig, 2003). Also, the objective function for this optimization may be non-convex, meaning that certain algorithms may become stuck in local minima. In many of these cases, where either the environment is stochastic or the search space is large, it is unlikely that an optimal plan can be found or will remain optimal throughout the mission. In addition, the definition of "optimal" in uncertain, dynamic, command and control environments may be difficult to quantify and represent in a single, static objective function.

Within the body of literature for task assignment algorithms, a wide variety of methods of solving the optimization have been developed, (see Clare, Cummings, & Bertuccelli, 2012 for a review). These methods include enumeration, the simplex method, dynamic programming,

36

branch and bound, and greedy algorithms. Meta-heuristic methods, often inspired by biological processes, include Genetic Algorithms, Simulated Annealing, Tabu Search, Particle Swarm Optimization, and Ant-Colony Algorithms, among others. Market-based auction algorithms are often applied to solve a variety of scheduling problems. Also, Dynamic Vehicle Routing (DVR) methods using Voronoi partitions are a potential solution method that could guarantee a certain level of performance without the need to replan constantly in a dynamic environment. DVR methods produce policies or decision rules, as opposed to specific task assignments, typically by optimizing the expected value of performance.

A recently developed method utilizes decentralized algorithms, which can solve the problem quickly with slightly sub-optimal solutions (Alighanbari & How, 2006; Choi, Kim, & Kim, 2011). Prior to the implementation of decentralized algorithms, a central node was used to collect information from all of the UVs, attempt to create a globally optimal schedule, and then send the plan back to all of the UVs. The drawbacks to these centralized scheduling algorithms are the high communication bandwidth necessary to collect global information, the increased computational resources necessary to plan for the entire team of UVs, and the vulnerability of the system to single node failures. In contrast, decentralized algorithms allow each UV to compute its locally best plan to accomplish the mission goals with shared information. Decentralized algorithms can potentially respond to changes in the environment more quickly, scale to larger numbers of UVs while taking advantage of each UV's added computational power, and are potentially more robust to communications failures (Alighanbari & How, 2006; Whitten, 2010). However, it can be difficult to reach a conflict-free schedule without needing a large amount of communication between the vehicles. In addition, the behavior of the UVs is emergent and sometimes difficult to predict in advance. Finally, decentralized algorithms cannot always guarantee optimal schedules.

This thesis will focus on human collaboration with a decentralized AS for real-time scheduling. In order to enable this collaboration, a *goal-based* architecture must be implemented (Clare & Cummings, 2011; Cummings, et al., 2012). The human operator guides the high-level goals of the team of UVs (as opposed to guiding each individual vehicle) and the AS assumes the bulk of computation for optimization of task assignments. The AS is responsible for decisions requiring rapid calculations or optimization, and the human operator supervises the AS for high-level goals

such as where to search and overall resource allocation (i.e., which tasks get included in the overall plan), as well as for tasks that require strict human approval, such as approving weapons release. The system could include the human operator, the graphical interface which displays information to the operator and allows the operator to interact with the system, the scheduling algorithm, and the semi-autonomous UVs which act in the environment, all with information flowing between components (Figure 3). It should be noted that the AS could exist as a stand-alone component, as pictured, or as sub-system of each UV, as in a decentralized system.
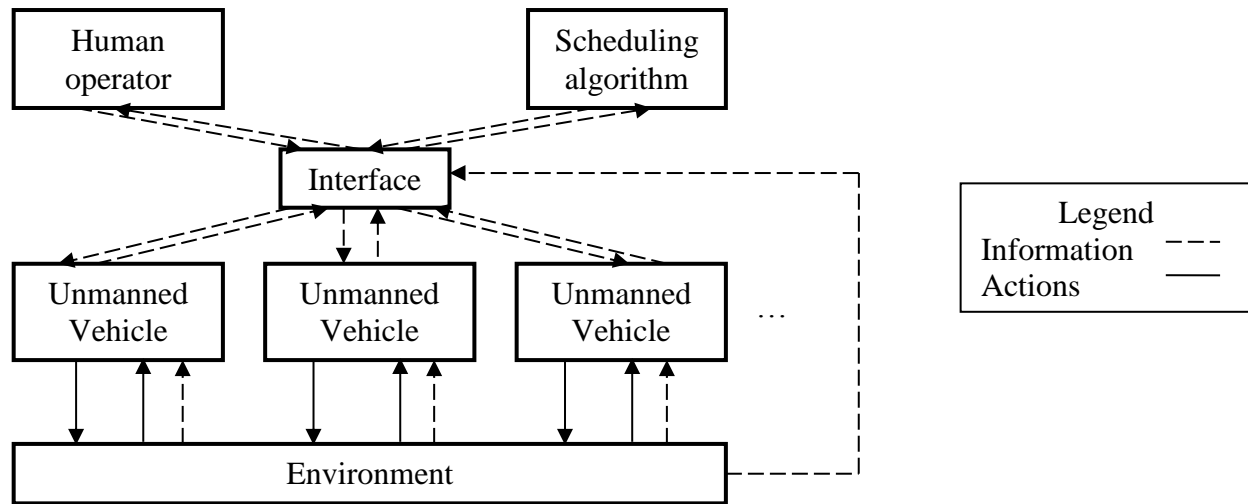
Figure 3. Human-automation collaborative scheduling system diagram.

This type of system has potential performance benefits for a realistic scenario with unknown variables, possibly inaccurate information, and dynamic environments by combining the computational ability of optimization algorithms with the judgment and adaptability of human supervisors (Cummings, et al., 2012). A *goal-based* architecture has been shown to achieve superior workload mitigation under high task loads with comparable system performance as compared to a vehicle-based, centralized control system (Clare & Cummings, 2011). Preventing high workload situations in a command and control environment is crucial for maintaining system performance, preventing costly or deadly errors, and enabling human operators to supervise larger fleets of coordinated UVs in the future.

In this architecture, the role of the human operator shifts from vehicle pilot to mission manager. This means that the human operator is monitoring the system and makes decisions to intervene throughout the mission to coach the automation. There are many possible role allocations for

operator and AS in the collaborative decision-making process (Bruni et al., 2007; Cummings & Bruni, 2009). However, due to the complexity of the scheduling problem, time-pressure, and the potential for cognitive overload (Cummings, Clare, et al., 2010; Wickens & Hollands, 2000), human operators in a *goal-based* architecture are not allowed to have vehicle-level control, i.e. directly control or task a single UV. The operator cannot take completely manual control and must work with the AS to develop plans for the UVs. The AS, however, cannot make changes to the strategic-level plan without the approval of the human operator and needs the human operator for other tasks (such as visual identification of targets). Thus, this system is truly collaborative, in that contributions from the human and automation are necessary to conduct the mission. The operator cannot take complete manual control or set the system to a purely automatic mode. This is a subtle, yet important distinction between previous research into human supervisory control of multiple UVs, where often the level of automation could be changed (Goodrich et al., 2001; Miller et al., 2005), or where the human could have direct control over an individual UV (Cummings, Nehme, & Crandall, 2007; Nehme, 2009).

The next section will review previously developed relevant models of humans in scheduling and control situations, in order to evaluate their applicability to real-time human-automation collaborative scheduling of multiple UVs, based on the above definition.

2.1 Previous Relevant Models

Computational models of human behavior in scheduling situations have existed since the early 1980s. In contrast to purely descriptive or conceptual models (Jackson, et al., 2004; Rad, 2008), computational/quantitative models typically leverage computer simulations to both promote deeper understanding of how human operators behave and provide testable predictions about human behavior under different circumstances (Gao & Lee, 2006; Parasuraman, 2000). One of the earliest computational models of human behavior in a scheduling situation was developed by Tulga and Sheridan (1980) to model human attention allocation methods during a multitasking supervisory control situation. Sanderson (1991) also developed a computational model of how humans perform scheduling functions in the manufacturing domain. In both of these early works, the impact of attention allocation strategies on operator workload and performance was shown. Also, Tulga and Sheridan (1980) developed a theory of the impact of the arrival rate of tasks on human cognitive workload. However, neither paper modeled the potential for human

collaboration with an algorithm for conducting these scheduling tasks, but instead focused on how humans conducted the scheduling process on their own.

With increases in computational power came an interest in pairing human operators with optimization algorithms to perform tasks that previously were done solely by a human or were impossible without automation. Thus, a set of computational models of human-automation collaboration were developed for general decision support systems (Riley, 1989), but mostly for industrial purposes such as manufacturing plants (Khasawneh et al., 2003) or process control (Gao & Lee, 2006; Lee & Moray, 1994). While these models of industrial processes captured the need for effective human-automation collaboration and the impact of operator trust on reliance on the automation, the manufacturing/industrial domain is distinct from control of multiple heterogeneous UVs. In a command and control mission involving multiple UVs, the environment is more dynamic and unpredictable, often with greater time pressure for decisions and higher uncertainty about the state of the environment. Also, operators controlling multiple UVs will likely have numerous additional tasks to conduct, requiring rapid task-switching, such as dealing with automation alerts, identifying visual imagery, monitoring the health and status of the UVs, and communicating with other operators and supervisors.

With the explosion in UV usage in the late 1990s and 2000s, the concept of a human operator controlling multiple UVs with the aid of an AS was explored. Given the differences between industrial processes and controlling UVs, a number of computational models have been developed specifically for human-automation collaboration involving multiple UV control. It should be noted that formal cognitive modeling techniques, such as Goals, Operations, Methods, and Selection (GOMS) rules (Card, Moran, & Newell, 1983) and Adaptive Control of Thought-Rational (ACT-R) (Anderson, 2007) have been applied to modeling human control of UVs (i.e. (Ball et al., 2002; de Visser, Jacobs, et al., 2012; Dimperio, Gunzelmann, & Harris, 2008; Drury, Scholtz, & Kieras, 2007)). However, these models focus on lower-level perception, cognition, and action processes, using a framework of discrete actions for visual encoding, memory access, memory retrieval, and motor actions. While formal cognitive models can produce accurate predictions of the time that an expert user would take to interact with an automated system, they often do not take into account variation in human operators, including the behavior of non-expert users, the impact of fatigue, previous operator experiences with automation, or the possibility of

40

operator mistakes or errors. All of these characteristics influence human decision-making and trust in real-time human-automation collaborative scheduling of multiple UVs. Finally, the focus of this thesis is on higher-level decision-making processes for human-automation collaboration. Human judgment and adaptability under uncertain conditions requires knowledge-based reasoning, rather than the skill- or rule-based decisions that formal cognitive models typically describe (Rasmussen, 1976, 1983).

Among the previously developed computational models which focus on higher-level decision-making processes (Cummings & Mitchell, 2008; Mkrtchyan, 2011; Nehme, 2009; Olsen & Wood, 2004; Rodas, Veronda, & Szatkowski, 2011; Savla et al., 2008), there are several crucial gaps with regards to real-time human-automation collaborative scheduling of multiple UVs, which will be discussed next.

2.1.1 Lack of Feedback Interaction among Important Aspects

The first limitation of the previous models is that they fail to effectively capture the feedback interactions among important aspects of real-time human-automation collaborative scheduling of multiple UVs. Feedback interactions play a crucial role in the collaboration between the human and automation in real-time scheduling of UVs. For example, as operator workload reaches too high of a level, a lack of attentional resources can lead to poor Situation Awareness (SA), defined as the "perception of the elements in the environment within a volume of time and space, comprehension of their meaning, and the projection of their status in the near future" (Endsley, 1995, p. 36). Inadequate SA can prevent the operator from properly updating his or her expectations about system performance or can lead to poor decision-making about when to intervene in the system, both of which can lead to poor overall performance and lower trust in the AS. Poor system performance and undertrust in the AS can cause the operator to intervene too rapidly, driving workload to an even higher level, thus reinforcing the feedback loop. Thus capturing all of these components in sufficient detail, along with the feedback relationships among them, is crucial to an effective model of human-automation collaborative scheduling of multiple UVs.

2.1.2 Lack of Explicit Contributions of Human and Automation

The second limitation of the previous models is that they do not capture sufficient details of the contribution of the operator and the AS to system performance. One of the reasons for this gap is that most of the previous models related to controlling multiple UVs (Cummings & Mitchell, 2008; Mkrtchyan, 2011; Nehme, 2009; Olsen & Wood, 2004; Rodas, et al., 2011; Savla, et al., 2008) assume that the human operator is directly individually tasking each vehicle, as opposed to *goal-based* control, where the human operator cannot command individual vehicles, but must guide the AS to create a viable schedule for all vehicles (Clare & Cummings, 2011). An effective model of human-automation collaborative scheduling must capture both the automation contribution to performance as well as the relationship between human guidance and system performance. The benefits of effective collaboration between the operator and an algorithm have been shown in previous research (Cummings, et al., 2012; Layton, et al., 1994; Silverman, 1992).

As explained earlier, *goal-based* architectures will likely use decentralized or non-deterministic algorithms. Decentralized control architectures enable each vehicle to compute its locally best plan to accomplish the mission goals with shared information (Choi, Brunet, & How, 2009). Non-deterministic algorithms have been utilized successfully for complex real-time scheduling problems, including genetic algorithms (Eun & Bang, 2007) and particle swarm optimization (Sujit, George, & Beard, 2008). While there are potential advantages to using decentralized algorithms or non-deterministic algorithms, an open question is how human operators will react to the unpredictability of working with these types of algorithms, where behavior is emergent and solutions are likely suboptimal, but generated quickly. For example, Walker et al. (2012) defined the concept of "neglect benevolence" to capture the idea that humans may need to allow time for decentralized swarm algorithms to stabilize before acting, and thus neglecting the swarm for a period of time may be beneficial to system performance. Capturing the impact of these and other AS characteristics (i.e. time to generate a plan, impact of operator interventions) would enable the model to predict system performance when working with different algorithms.

2.1.3 Lack of Integration of Qualitative Variables

Third, none of the models which are specifically for multi-UV control capture the influence of qualitative variables such as trust or alignment of human and AS goals on system performance.

Operator trust in the AS can change throughout a mission due to the operator's perception of how well the AS is performing towards the operator's goals. Both overtrust and undertrust can have a significantly negative impact on system performance for human-automation collaborative scheduling. In real-time multiple-UV scheduling, the mission may be so complex that the operator cannot bypass the AS without the operator's workload reaching levels that would cause a decrease in performance (Cummings & Nehme, 2010). This requirement to continue working with the AS means that it is critical for an effective model to capture the dynamics of trust and the impact of human and AS goal alignment.

2.1.4 Summary

Three gaps with regards to real-time human-automation collaborative scheduling of multiple UVs have been identified among the previously developed models reviewed above. First, none of the previous models capture the feedback interactions among important aspects of a real-time human-automation collaborative scheduling system. Capturing these feedback interactions is necessary for an effective model of a *goal-based* architecture for controlling multiple UVs. Second, the relevant models do not capture sufficient details of the AS in order to model the automation contribution to performance as well as the relationship between human guidance and system performance. Third, none of the models which are specifically for multi-UV control capture the influence of qualitative variables such as trust or alignment of human and AS goals on system performance.

This thesis proposes to address these three gaps through the development of a computational model of real-time human-automation collaborative scheduling of multiple UVs. This new model will aid designers in dealing with the previously discussed issues of AS brittleness, inappropriate levels of operator trust, high operator workload, and poor goal alignment between the human and AS. To inform the model development process, the next section identifies attributes that are important to consider when modeling real-time human-automation collaborative scheduling.

2.2 Model Attributes

There is a large body of literature exploring the various features of human supervisory control, especially for UV control (i.e. (Chen, Barnes, & Harper-Sciarini, 2011; Crandall & Cummings, 2007; Cummings, Bruni, & Mitchell, 2010; Dixon & Wickens, 2003; Endsley, 1995; Lee & See,

2004; Miller, 2004; Nehme, 2009; Sanders et al., 2011; Sheridan, 1992)). Based on the previous literature and the above definition, it is proposed that a model of real-time human-automation collaborative scheduling of multiple UVs should capture the following attributes:

- Attention allocation and situation awareness
- Cognitive workload
- Trust in automation
- Human learning
- Automation characteristics
- Human value-added to performance through interventions
- Team coordination and structure (not addressed in this thesis)

In the following sections, it is argued that these attributes collectively capture the major human, automation, and combined human/automation performance issues associated with human-automation collaborative scheduling of multiple UVs and that all of these attributes are necessary in an effective model of such a system. The discussion of these attributes focuses primarily on their importance to the domain of this thesis, multiple UV scheduling, although the extension of this model to scheduling domains outside of multiple UVs is explored in Chapter 6. This thesis focuses on the concept of a single operator controlling multiple UVs, thus team coordination and task allocation among multiple operators is not considered and is left for future work. In the following sub-sections, for each of the remaining six attributes, previous research will be presented supporting the importance of capturing the attribute when modeling human-automation collaborative scheduling of multiple UVs. While the sub-sections will discuss each attribute independently, the interactions between the various attributes are important and will be discussed further in Chapter 3.

2.2.1 Attention Allocation and Situation Awareness

Human-automation collaborative scheduling of multiple UVs inherently involves multi-tasking, as operators must pay attention to the activities of multiple vehicles. In addition, operators engaged in these dynamic, high workload environments must both concentrate attention on the primary task (e.g., monitoring vehicle progress and identifying targets) and also be prepared for various alerts, including incoming chat messages or automation notifications about potential

changes to the vehicle schedules. This need to concentrate on a task, yet maintain a level of attention for alerts requires both interrupt and task-driven processing. The allocation of attention between these two can incur cognitive costs that negatively impact overall system performance (Miyata & Norman, 1986). Poor attention allocation has been shown to be a significant contributor to poor operator performance in single operator control of multiple unmanned vehicles (Crandall & Cummings, 2007; Goodrich, Quigley, & Cosenzo, 2005). Thus, capturing how the operator allocates his or her attentional resources (Wickens & Hollands, 2000) and the switching costs (Miyata & Norman, 1986) involved in multi-tasking are both important to the model proposed in this thesis.

The result of poor attention allocation and information processing efficiency can be low Situation Awareness (SA), which can decrease the effectiveness of an operator's decisions. Maintaining adequate SA of both the overall mission as well as individual UVs has been described as "one of the most critical factors for achieving effective supervisory control of multiple UVs" (Chen, et al., 2011, p. 439). There are three levels of SA: 1) perception of the elements in the environment within a volume of time and space, 2) comprehension of their meaning, and 3) the projection of their status in the near future (Endsley, 1995). There is a clear relationship between attention allocation and SA, as it has been shown in previous studies that switching between tasks incurs a cost in terms of SA, leading to delays in responses and errors (Cummings & Mitchell, 2008; Squire, Trafton, & Parasuraman, 2006).

While all three levels of SA are generally important, Level I SA, perception of changes in the environment and of changes in system performance, is especially crucial to human-automation collaborative scheduling of multiple UVs. As described previously, the human operator must decide when to intervene to create tasks, change the schedule, or modify the way the AS works. This decision to intervene is driven by the operator's timely perception of changes to the environment or mission performance, thus it will be important to the proposed model to capture time delays in the perception of these changes.

2.2.2 Human Cognitive Workload

Human cognitive workload is defined as the mental resource demand experienced by the operator as a function of task load, or the level of tasking that an operator is asked to perform

(Wickens & Hollands, 2000). As theorized by the Yerkes Dodson Law (1908), up to a certain point, increased workload can be beneficial to performance. Once the operator reaches what will be referred to as "cognitive overload," performance begins to suffer.

It has been shown in numerous previous empirical studies that cognitive workload has a significant impact on both human and system performance in human supervisory control of multiple UVs (Clare & Cummings, 2011; Cummings, Clare, et al., 2010; Cummings & Guerlain, 2007; Cummings & Nehme, 2010; Dixon & Wickens, 2003; Nehme, 2009; Rouse, 1983; Ruff, Narayanan, & Draper, 2002; Schmidt, 1978). Thus it will be crucial in the proposed model to represent both the potential benefits of increased workload and the detrimental effects of cognitive overload. Beyond the traditional measures of workload through subjective ratings (Hart, 1988; Rubio et al., 2004), objective measures of workload such as utilization, the ratio of the total operator "busy time" to the total mission time, have been successfully incorporated in previous models of operator workload (Nehme, 2009; Schmidt, 1978) and should be incorporated in the model presented in this thesis.

It should be noted that this thesis will only focus on medium to high task load environments, while low task load conditions and the impact of fatigue and boredom have and continue to be explored by others (Cummings, et al., 2013; Mkrtchyan, 2011; Scerbo, 2001; Walters, et al., 2000). This is another area of possible future work.

2.2.3 Human Trust in Automation

Human trust in the AS is a crucial driver of performance in a human-automation collaborative scheduling system. Although there are some similarities to the concept of trust between two humans, there are also some significant differences between human-human trust and human-automation trust (Muir, 1987). Human trust in an AS can be defined as the "attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability" (Lee & See, 2004, p. 51). This thesis distinguishes trust in a scheduling algorithm for controlling UVs from recent research on human trust in embodied agents (de Visser, Krueger, et al., 2012) or robots (Hancock et al., 2011), as the operator is not co-located with the vehicle and there is no physical embodiment of the algorithm. Operator trust in the AS can fluctuate due both to the operator's initial trust level in the AS and the behavior of the AS throughout the

46

mission. This phenomenon has been observed in data analysis from a previous human-in-the-loop experiment and has been linked to changes in performance (Clare, Macbeth, et al., 2012; Gao, et al., 2013).

Both overtrust and undertrust in automation can be detrimental to system performance. Low human trust in the AS can be caused by automation "brittleness," in that the AS can only take into account those quantifiable variables, parameters, objectives, and constraints identified in the design stages that were deemed to be critical (Scott, et al., 2002; Silverman, 1992; Smith, et al., 1997). In a command and control situation such as supervising multiple UVs, where unanticipated events such as weather changes, vehicle failures, unexpected target movements, and new mission objectives often occur, AS have difficulty accounting for and responding to unforeseen changes in the environment (Guerlain, 1995; Polson & Smith, 1999). Additionally, the designers of optimization algorithms often make a variety of assumptions when formulating the optimization problem, determining what information to take into account, or, in the case of receding horizon algorithms, deciding how far into the future to plan (Bellingham, et al., 2002; Layton, et al., 1994). Operators with low trust may spend an excessive amount of time replanning or adjusting the schedule (Clare, Macbeth, et al., 2012; Cummings, Clare, et al., 2010).

Also, "overtrust" in automation has been cited in a number of costly and deadly accidents in a variety of domains (Cummings, 2004a; Parasuraman & Riley, 1997). Overtrust in the AS can lead to the phenomenon of automation bias (Mosier et al., 1998), where operators disregard or do not search for contradictory information in light of an AS-generated solution which is accepted as correct (Cummings, 2004a). A number of empirical studies have shown that when working with imperfect automation, automation bias occurs (Chen & Terrence, 2009; Lee & Moray, 1994; Muir & Moray, 1996; See, 2002).

Thus, achieving the "appropriate" level of trust, calibrated to the reliability of the AS under different situations, is essential to achieving high performance in a human-automation collaborative scheduling system (Lee & See, 2004). To effectively model the trust calibration process, it is essential to start by capturing the operator's initial trust level, which can vary widely based on the operator's prior knowledge, past experiences, and training (Lee & Moray,

1994; Moray, Inagaki, & Itoh, 2000). Trust is dynamic, however, and can fluctuate throughout a mission based on the operator's perception of how well the AS is performing (Lee & Moray, 1992; Muir & Moray, 1996). A number of studies have found that human trust has inertia, where automation errors do not necessarily cause instantaneous loss in trust, but recovery in trust from severe failures can also be slow (Hoffman et al., 2013; Lee & Moray, 1992, 1994; Lewandowsky, Mundy, & Tan, 2000; Parasuraman, 1993; See, 2002). Capturing the impact of this inertia on the dynamics of the trust calibration process is also important to the model proposed in this thesis.

Finally, it should be noted that previous studies and models of trust focused on systems where the operator had a choice between automated control or turning off the automation and taking complete manual control based on the difference between the operator's trust in the automation and self-confidence (Gao & Lee, 2006; Lee & Moray, 1992). Based on the definition of human-automation collaborative scheduling of multiple UVs presented earlier in this Chapter, however, taking complete manual control is not feasible. Thus, while the model's treatment of trust will be derived from previous models of trust, it will require a slightly different approach that focuses on when and how the operator decides to intervene to guide the system, not when the operator decides to take complete manual control because his or her self-confidence in system operation is higher than his or her trust in the automation.

2.2.4 Human Learning

As human operators gain experience with a human-automation collaborative scheduling system, they will both learn how to use the system more quickly and efficiently and learn how to better collaborate with the AS. Sheridan (2006, p. 1028) stated that in a human supervisory control system, one of the operator's major roles is "learning from experience so as to do better in the future." Even with adequate initial training, operators controlling multiple UVs in a dynamic and uncertain environment will likely encounter novel situations throughout a mission that provide the operator with new knowledge and experience.

For the proposed model in this thesis, there are two forms of learning that should be captured. The first, called "long-term learning" in this thesis, is the adjustment of the operator's expectations of system performance. Tversky and Kahneman (1974) explained that humans who

48

need to estimate a value often utilize an "anchoring and adjustment" heuristic, where they begin with an initial guess and then attempt to adjust the guess based on new information gathered. In human-automation collaborative scheduling of multiple UVs, the operator will likely begin with a certain expectation of how well the system should perform a given task. Based on the operator's perception of system performance throughout the mission, the operator will adjust his or her expectation of performance, albeit with some amount of "inertia" or a time delay in the adjustment (Lee & See, 2004). Representing this change in the operator's expectations throughout the mission is important to the proposed model because previous research has shown that people (and businesses) often decide to make changes when there is a large enough difference between expected performance and actual (or perceived) performance (Rudolph, Morrison, & Carroll, 2009; Sastry, 1995; Tushman & Romanelli, 1985).

The second form of learning, called "short-term learning" in this thesis, is the learning curve of how to use the graphical user interface more efficiently and how to read and analyze the information presented in the various displays more quickly. Many of the empirical data sets that can be used for verification and validation of the model proposed in this thesis utilize novice operators who receive some limited training but, in fact, are still learning how to use the system during experimental trials. Evidence of learning in experiments involving humans controlling multiple UVs has been found in previous studies (Cooper & Goodrich, 2008; Mekdeci & Cummings, 2009). The concept of a learning curve was first coined to describe the rate of increase in the productivity of airplane manufacturing workers (Wright, 1936). Since then, the concept has been well explored in the psychology, manufacturing, and business domains.

Finally, it should be noted that others are researching machine learning (Geramifard et al., 2012; Michini, Cutler, & How, 2013) and adaptive automation techniques (de Visser, Jacobs, et al., 2012; Miller, et al., 2005) to enable the automation for controlling UVs to learn over time as well. This thesis focuses on human learning and the incorporation of automation learning into a model of real-time human-automation collaborative scheduling is left for future work.

2.2.5 Automation Characteristics

In a model of human-automation collaborative scheduling of multiple UVs, certain characteristics of the AS must be captured in order to effectively model the system. For example,

it has been shown in previous research that the rate at which the AS prompts the operator to approve new schedules impacts operator workload and system performance (Clare, Maere, & Cummings, 2012; Cummings, Clare, et al., 2010). A prior empirical study showed that providing the operator with an AS that allows modifications to the objective function of the algorithm can improve SA, lower workload, and improve the operator's perception of the AS (Clare, Cummings, How, et al., 2012). Lee and See (2004) identified a number of the bases of trust in automation, among them predictability, the degree to which future behavior can be anticipated; dependability, the degree to which behavior is consistent; and reliability, a measure of how well the automation performs relative to current and historical performance. Other factors that can influence both human and system performance include the speed with which the AS can produce new schedules and the alignment of objectives between the AS and the operator (Howe, et al., 2000; Linegang et al., 2006; Silverman, 1992).

As described earlier in this Chapter, a wide variety of AS have been developed for assigning tasks to multiple UVs in real-time, with differences in their solution method, guarantees about optimality, whether or not the AS took into account uncertainty, whether the AS was centralized or distributed among the UVs, and whether the AS could plan for heterogeneous UVs (Clare, Cummings, & Bertuccelli, 2012). Thus, for the model proposed in this thesis to be effective and capable of evaluating the impact of different AS on human and system performance, the impact of such AS characteristics on performance must be captured.

2.2.6 Human Value-Added Through Interventions

A number of studies have shown that humans collaborating with algorithms can achieve higher performance than either the human or the algorithm alone under certain conditions (Anderson, et al., 2000; Cummings, et al., 2012; Cummings & Thornburg, 2011; Johnson, et al., 2002; Malasky, et al., 2005; Ponda, et al., 2011; Ryan, 2011). Scott, Lesh, and Klau (2002) showed that in experiments with humans utilizing mixed-initiative systems for vehicle routing, operator intervention can lead to better results, but there is variation in the way that operators interact with the system and in their success in working with the automation. Cummings, et al. (2012) showed that in a human-automation collaborative scheduling system for multiple UVs, motivated human operators clearly added value to the performance of the system as compared to the performance of the system without a human actively guiding the system.

50

Thus, the proposed model should both capture baseline system performance without significant human intervention and identify the areas where humans could add the most value. Once those areas are identified, it is important to then quantify the impact of the different interventions that the human operator can undertake. Rather than a standalone variable, the model can represent human value-added through a feedback process that combines the impact of operator interventions with the impact of operator cognitive workload. The rate of operator interventions and the effectiveness of these interventions can form competing feedback loops (Rudolph & Repenning, 2002), as previous empirical research has shown that rapid rates of replanning can cause an increase in workload and a decrease in system performance (Clare & Cummings, 2011; Cummings, Clare, et al., 2010).

2.3 Chapter Summary

In summary, the concept of real-time human-automation collaborative scheduling of multiple UVs has been defined. The representative setting for this thesis is a reconnaissance mission to search for an unknown number of mobile targets. The mission scenario is multi-objective, and includes finding as many targets as possible, tracking already-found targets, and neutralizing all hostile targets. Scheduling is defined here as creating a temporal plan that assigns tasks/targets among the team of heterogeneous UVs, determines when the tasks will be completed, and takes into account capability, location, and timing constraints. In order to conduct this mission in uncertain, dynamic environments, a human operator will collaborate with a decentralized AS through a *goal-based* architecture to guide the team of UVs.

Based on this definition, a review of previously developed relevant models of humans in scheduling and control situations was conducted to evaluate their applicability to the representative setting. The review identified crucial gaps with regards to real-time human-automation collaborative scheduling of multiple UVs. Previous computational models did not capture the feedback interactions among important aspects of human-automation collaboration, the impact of AS characteristics on the contributions of the operator and automation to system performance, and the impact of qualitative variables such as trust in automated scheduling algorithms.

Finally, six attributes that are important to consider when modeling real-time human-automation collaborative scheduling are proposed, providing a theoretical basis for the model proposed in this thesis. Attention allocation and situation awareness, cognitive workload, trust in automation, human learning, automation characteristics, and human value-added through interventions should all be captured by the proposed model.

Chapter 3 describes the modeling process that created the Collaborative Human-Automation Scheduling (CHAS) model, a System Dynamics (SD) model of human-automation collaborative scheduling of multiple UVs. The model is described in detail, including how it captures the important attributes identified in this chapter and addresses the gaps in previous models. Then, Chapters 4 and 5 describe preliminary validation of the model using historical data sets and a new human subject experiment.

3 Model Development

This chapter describes the modeling process that created the Collaborative Human-Automation Scheduling (CHAS) model, a System Dynamics (SD) model of human-automation collaborative scheduling of multiple Unmanned Vehicles (UVs). The chapter begins by describing the field of System Dynamics and why it is appropriate for modeling a human-automation collaborative scheduling system. Next, the model building process is described, starting with an analysis of a previous experimental data set where a human operator was paired with an Automated Scheduler (AS) to conduct collaborative scheduling of multiple UVs in a simulation testbed. This analysis identified reference modes of operator behavior and performance, leading to the creation of a "dynamic hypothesis." This hypothesis attempts to explain the dynamics of the system as endogenous consequences of the feedback structure of this system. Next, a SD model was formulated from this hypothesis. Parameters, behavioral relationships, and initial conditions of the system were estimated from experimental data. The chapter concludes by describing the outputs and benefits of the model.

3.1 System Dynamics Modeling

System Dynamics (SD) is a well-established field that draws inspiration from basic feedback control principles to create simulation models (Sterman, 2000). SD constructs (stocks, flows, causal loops, time delays, feedback interactions) enable investigators to describe and potentially predict complex system performance, which would otherwise be impossible through analytical methods (Forrester, 1961). SD models have been used in a number of large and small scale systems with social elements including management, economics, logistics, education, and disease spread (Sterman, 2000). More relevant to real-time human-automation collaborative scheduling, SD models have been developed to represent human supervisors monitoring automated systems (White, 2003), a number of command and control applications (Coyle, Exelby, & Holt, 1999), human decision-making in high-stress situation with interruptions (Rudolph & Repenning, 2002), and human problem-solving under time-pressure in action-oriented environments, such as doctors coping with an operating room emergency (Rudolph, et al., 2009). This proposed application of SD is novel because no previous efforts have used this

method to model human interaction with automated systems at the decision-making level or to represent UV systems.

While other simulation modeling techniques, such as Discrete Event Simulation (DES), have been successfully applied to modeling human supervisory control of multiple UVs (Nehme, 2009; Rodas, et al., 2011), there are a number of reasons that SD models are particularly appealing for UV systems. The first is the ability of SD models to capture non-linear processes (Sweetser, 1999). Since human performance does not generally adhere to linear models (Cummings, et al., 2013; Gao & Lee, 2006), using non-linear behavioral and performance representations will be critical for the external validity of the model. Second, SD models can include both qualitative and quantitative data (Özgün & Barlas, 2009; Sterman, 2000). As previously discussed in Chapter 2, capturing the dynamics of trust and the impact of human and AS goal alignment on system performance is crucial for an effective model. Third, SD models are effective at capturing the impact of latencies and feedback interactions on the system, which is essential for modeling a human operator and the impact of delays in perception of system performance on operator behavior and trust. While there is an ongoing argument in the modeling community about the scenarios for which SD is more appropriate than DES (i.e. (Özgün & Barlas, 2009; Sweetser, 1999)) a key contribution of this thesis is the evaluation of the adaptation of SD techniques to model human supervisory control of UVs in comparison to DES techniques.

The model presented in this thesis was developed through an inductive modeling process. Größler and Milling defined the inductive SD modeling process by stating that "the solution to a specific problem is sought as well as a specific situation serves as the basis for the model. Later in the process, insights gained in the project might be generalized…" (Größler & Milling, 2007, p. 2). The SD modeling process can be broken down into five major phases (Sterman, 2000). First, in the problem articulation stage, the overall problem that the model is attempting to represent is identified, along with key variables to be captured within the boundary of the model. Much of this work was described in Chapter 2. In the second stage, a "dynamic hypothesis" is developed. A dynamic hypothesis is defined as a theory that explains the behavior of the system as an endogenous consequence of the feedback structure of the holistic system (Sterman, 2000). It is a working hypothesis that guides the modeling effort and is continuously tested and refined

throughout the model building and testing process. Sections 3.2 and 3.3 describe the data analysis that led to the dynamic hypothesis for the model described in this thesis.

In the third stage, the dynamic hypothesis is mapped into causal loops and stocks and flows in order to formulate the simulation model and estimate exogenous parameters, as described in Section 3.4. The SD community defines endogenous variables simply as those variables which are calculated within the model, while exogenous variables are assumed parameters which lie outside of the model boundary (Sterman, 2000). The fourth stage, testing the model, including comparison of model outputs to experimental data sets, robustness under extreme conditions, sensitivity analyses and other tests are described in Chapters 4 and 5. The fifth stage, policy design and evaluation, including evaluating the ability of this model to predict performance under new circumstances, is described in Chapter 5. Finally, Chapter 6 will describe how this model, developed for a specific system, can be generalized for use with other real-time human-automation collaborative scheduling systems. While these phases will be described in a linear fashion in this thesis, the model underwent significant iteration, including a model reduction process (Appendix A). The final parsimonious model is described here.

3.2 Previous Experimental Data Set Analysis

In order to inform the construction of the CHAS model, a time-series data analysis was conducted using a previous experimental data set. First, the testbed used to collect this data is described. Then the analysis of the data is presented.

3.2.1 Testbed Description

The testbed which the CHAS model was designed to represent is a collaborative, multiple UV system called Onboard Planning System for UVs Supporting Expeditionary Reconnaissance and Surveillance (OPS-USERS), which leverages decentralized algorithms for vehicle routing and task allocation (Cummings, et al., 2012). This system functions as a computer simulation but also supports actual flight and ground capabilities (How et al., 2009); all the decision support displays described here have operated actual small air and ground UVs in real-time (Kopeikin et al., 2012).

Operators were placed in a simulated command center where they controlled multiple, heterogeneous UVs for the purpose of searching an area of interest for new targets, tracking these targets, and approving weapons launch for hostile targets. The UVs in the scenario included one fixed-wing UAV, one rotary-wing UAV, one Unmanned Surface Vehicle (USV) restricted to water environments, and a fixed-wing Weaponized Unmanned Aerial Vehicle (WUAV). The UAVs and USV were responsible for searching for targets, using a decentralized, local search algorithm to guide their search pattern (Whitten, 2010). Once a target was found, the operator was alerted to perform a target identification task (i.e., hostile, unknown, or friendly), along with assigning an associated priority level (i.e., high, medium, low). Then, hostile targets were tracked by one or more of the vehicles until the human operator approved WUAV missile launches. UVs automatically returned to a central base when they needed to refuel. A primary assumption was that operators had minimal time to interact with the displays due to other mission-related tasks.

Participants interacted with the OPS-USERS simulation via two displays. The primary interface is a map display (Figure 4). The map shows both geo-spatial and temporal mission information (i.e., a timeline of mission significant events), and supports an instant messaging "chat" communication tool, which provides high level direction and intelligence. As in real-life scenarios, changing external conditions often require the human and the system to adapt, which are represented through "Rules of Engagement" (ROEs) received through the chat tool. Icons represent vehicles, targets of all types, and search tasks, and the symbology is consistent with MIL-STD 2525 (U.S. Department of Defense, 1999). Operators had two exclusive tasks that could not be performed by automation: target identification and approval of all WUAV weapon launches. Operators could also create search tasks, which dictated on the map those areas which the operator wanted the UVs to specifically search.

The AS used in OPS-USERS is the Consensus Based Bundle Algorithm (CBBA), a decentralized, polynomial-time, market based protocol (Choi, et al., 2009). More details on the AS can be found in (Whitten, 2010), with details of the OPS-USERS automation architecture in (Clare, Cummings, How, et al., 2012; Cummings, et al., 2012). The performance plot (Figure 4) gives operators insight into the AS performance, as the graph shows predicted plan score (red) versus current plan score (blue) of the system. The AS calculates plan score based on an

56

objective function that the operator can modify, as described below. When the AS generates a new plan that is at least five percent "better" than the current plan, the Replan button turns green and flashes, and a "Replan" auditory alert is played. The operator can choose to replan at any time, regardless of whether the Replan button is flashing. When the Replan button is clicked, the operator is taken to the Schedule Comparison Tool (SCT), for conducting scheduling tasks in collaboration with the automation.
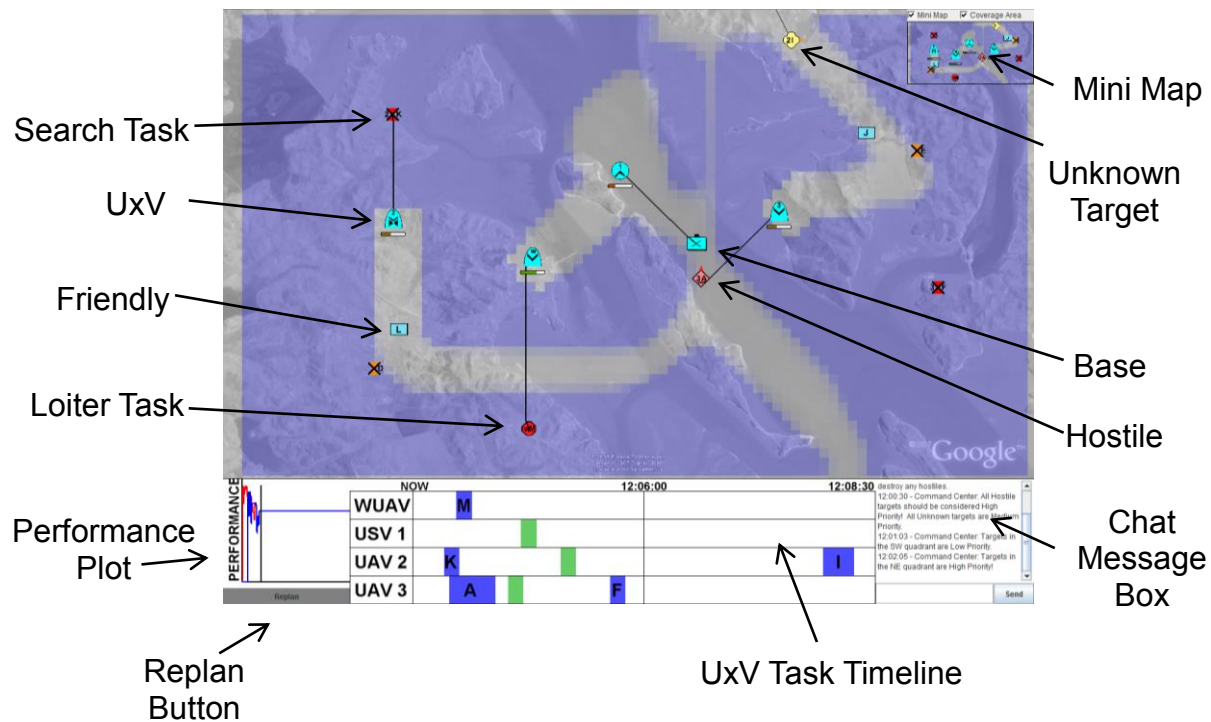
Figure 4. Map Display.

The SCT display (Figure 5) appears when the Replan button is pressed, showing three geometrical forms colored gray, blue, and green at the top of the display, which are configural displays that enable quick comparison of schedules. The left form (gray) is the current UV schedule. The right form (green) is the latest automation-proposed schedule. The middle working schedule (blue) is the schedule that results from user plan modification. The rectangular grid on the upper half of each shape represents the estimated area of the map that the UVs will search according to the proposed plan. The hierarchical priority ladders show the percentage of tasks assigned in high, medium, and low priority levels, respectively.

When the operator first enters the SCT, the working schedule is identical to the proposed schedule. The operator can conduct a "what-if" assignment by dragging the desired unassigned tasks into the large center triangle. This query forces the automation to generate a new plan if possible, which becomes the working schedule. The configural display of the working schedule alters to reflect these changes. However, due to resource shortages, it is possible that not all tasks can be assigned to the UVs, which is representative of real world constraints. The working schedule configural display updates with every individual query so that the operator can leverage direct-perception interaction (Gibson, 1979) to quickly compare the three schedules. This "what-if" assignment, which essentially is a preview display (Wickens & Hollands, 2000), represents a collaborative effort between the human and automation (Layton, et al., 1994). Operators adjust team coordination metrics at the task level as opposed to the individual vehicle level, which has been shown to improve single operator control of a small number of multiple, independent robots (Goodrich et al., 2007). Operators could also modify the objective function that the AS uses to evaluate schedules for the UVs through the "Plan Priorities" panel on the right side of the SCT. Details of the OPS-USERS interface design and usability testing can be found in (Clare, Cummings, How, et al., 2012; 2008).
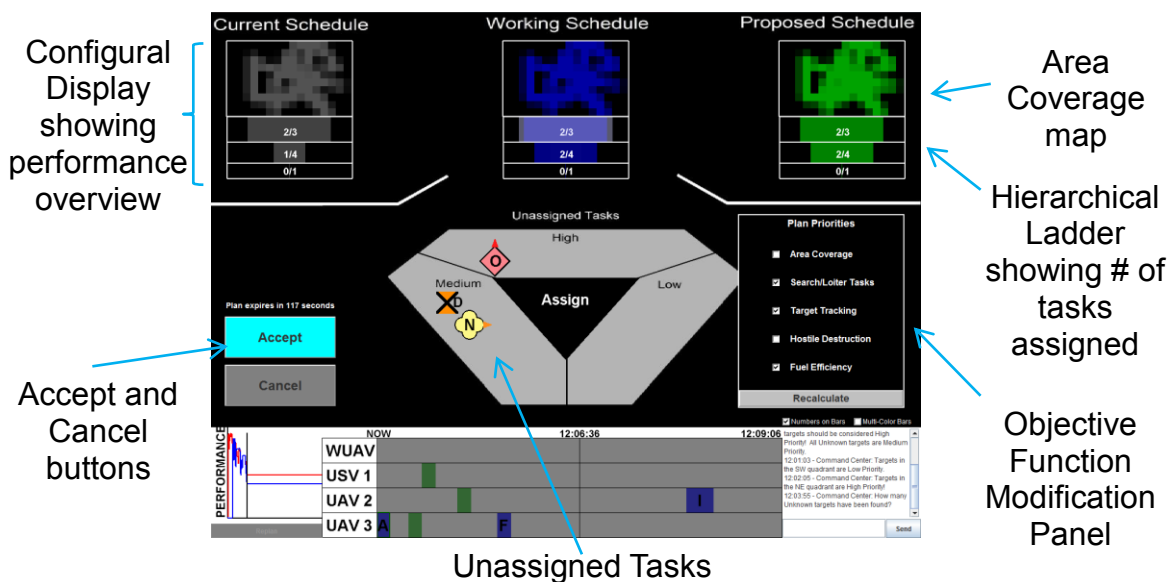
Figure 5. Schedule Comparison Tool (SCT)

Operators can either choose to accept the working schedule or cancel to keep the current schedule. Upon accepting a new schedule, the AS only communicates to the vehicles via a

prioritized task list, and the vehicles sort out the actual assignments amongst themselves. This human-automation interaction scheme is one of high level *goal-based* control, as opposed to more low-level vehicle-based control.

To summarize, there were four possible categories of "interventions" that the human operator could perform to adjust how the teams of UVs conducted the mission. First, the operator could create search tasks, which enabled the human operator to guide the automation and prevent myopic behaviors by encouraging the UVs to search in remote, unsearched areas that were likely to contain new targets. Second, the operator could "replan" by asking the AS to generate a new schedule for the team of UVs. This new schedule must be approved by the operator before the team of UVs will enact the new schedule. Third, the operator could modify the objective function of the AS throughout the mission to ensure that the goals of the AS align with changing mission goals. Fourth, the operator could conduct a "what-if assignment" to attempt to manually force a single task into the schedule if possible and view the ramifications of this forced assignment.

3.2.2 Data Analysis

The data set for this analysis is from an experiment where 30 participants performed two 20-minute long simulated UV missions (Clare, Cummings, How, et al., 2012). Each scenario had 10 targets initially hidden to the operator. Rules of Engagement (ROEs) were provided to the operators every 5 minutes via a chat box, which adjusted the goals that the operator focused on during each phase of the mission. It was assumed that all UVs and sensors operated normally throughout the mission.

The primary mission performance metrics collected in these experiments were percentage of area covered during the search process and percentage of targets found, which were logged once per minute. All other metrics were collected in two minute intervals, resulting in 10 data points for each mission that enable comparison and aggregation across trials. These other metrics included: length of time spent replanning, utilization (a proxy measure for cognitive workload), the probability of performing a what-if assignment, the probability of modifying the objective function of the AS, the number of replans per two minute interval, and the number of search

tasks created per two minute interval. Aggregate data for all 60 trials in the experiment are shown in Figure 6.
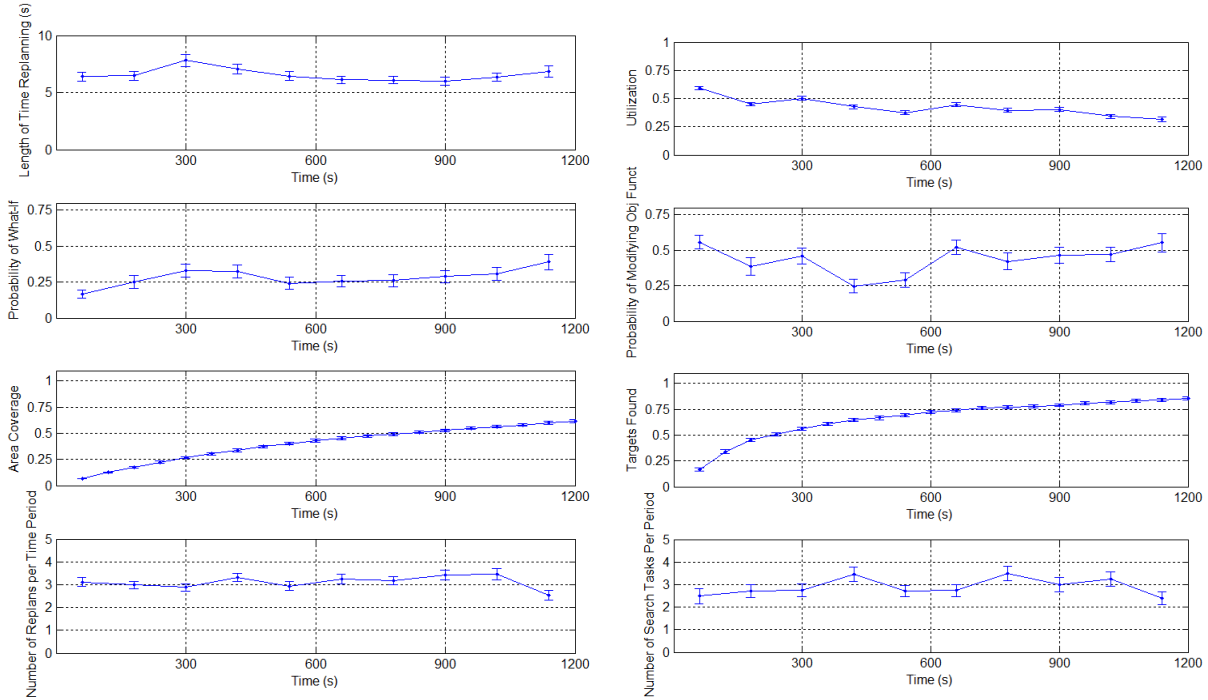
Figure 6. Aggregate time series experimental data. Standard Error bars are shown.

A common analysis in the SD modeling process is to look for reference modes, which are graphs and data showing changes in the system over time. A few key reference modes that are apparent from these plots include that utilization, the percent of the overall mission time an operator was engaged in a task, generally declined throughout the mission, from 60% busy time to only 30% busy time by the end of the mission. Also, at the start of the mission, operators only performed a what-if assignment with 17% of the schedules proposed by the AS. After 5 minutes into the mission, operators approached a 33% likelihood of conducting a what-if assignment, which remained roughly flat for the rest of the mission. Next, it appears that operators generally had about a 50% likelihood of making a modification to the objective function of the AS, except during the period of the mission from 300-600 seconds, when that likelihood dropped to roughly 25%. Finally, two of the key performance metrics, area coverage and targets found, both roughly resemble saturation curves, where the initial rates of covering area and finding new targets are high at the start of the mission, but later taper off as there is less new area to cover and fewer new targets left to be found.

60

As one of the key goals of this modeling effort is to capture the drivers of system performance, a cluster analysis was conducted to identify the missions which had significantly high or low performance so that they could be analyzed. Two separate clustering analyses were conducted, with the first analysis using total area coverage by the end of the mission as the clustering metric, and the second analysis using total number of targets found by the end of the mission. A hierarchical clustering was conducted using Ward's Method to determine the number of clusters. Afterwards, the k-means algorithm was used to assign missions to clusters. Following clustering, operator behavior (search task creation rates, workload, length of time replanning, frequency of replanning, etc.) was compared between the high and low performance clusters for each performance metric. Between the best and worst missions ranked in terms of targets found, there were no significant differences found in operator action measures (Appendix B).

There were significant differences in operator behavior, however, for the best and worst missions ranked in terms of area coverage, so data analysis focused on this performance metric. The area coverage metric of performance also provides a number of beneficial features to this analysis and modeling effort. It is a continuous measure of performance as opposed to discrete measures such as targets found or hostiles destroyed. It is visible to the human operator due to the use of a "Fog of War" overlay on the Map View (Figure 4). Also, it has been shown in prior work that the human operator can make a significant and meaningful contribution to the collaborative relationship with the AS in terms of area coverage performance (Cummings, et al., 2012).

The clustering analysis results based on the total area coverage in each mission are shown in Figure 7a. The red circled clusters were identified as the Low Performance and High Performance clusters. Of the total 60 missions, there were 11 missions in the High Performance cluster and 26 missions in the Low Performance cluster. The average area coverage performance over time through the mission is shown for the two clusters in Figure 7b. A repeated measures ANOVA showed a significant difference between the two groups in terms of area coverage, $F(1,35) = 145.801$, $p < 0.001$.
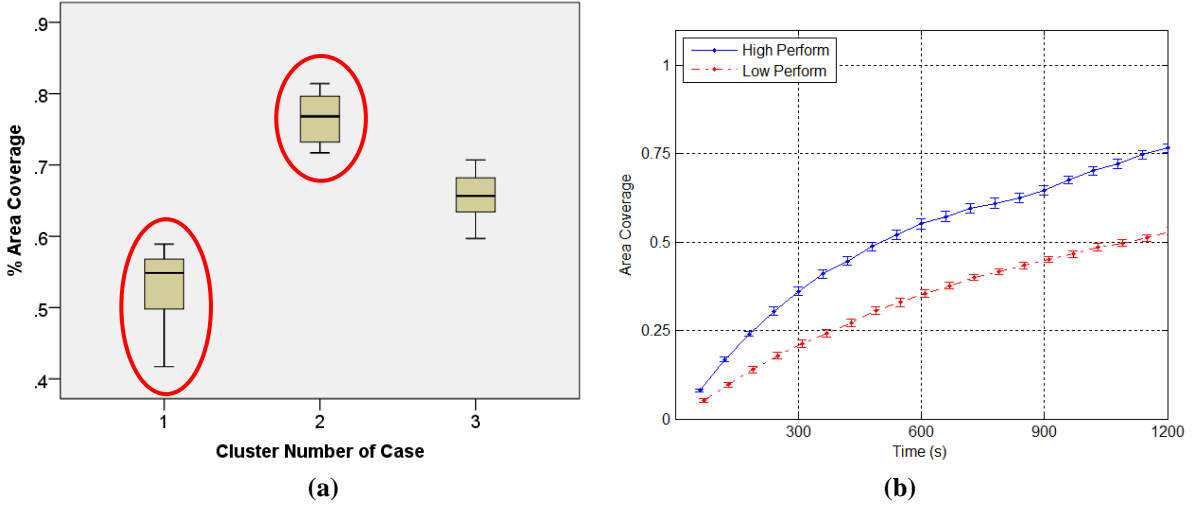
Figure 7. (a) Results of cluster analysis using total area coverage as the clustering metric. (b) Average area coverage performance over time for the high and low performance clusters. Standard Error bars are shown.

Next a comparison of operator actions between the two clusters was conducted using a repeated measures ANOVA. The full details of this analysis are presented in Appendix B. The operator actions where there was a significant difference ($\alpha=0.05$) between the two clusters are shown in Figure 8a-c. These operator actions were:

- The number of replans per 120-second time period, also known as the replan rate. High performers replanned more frequently than low performers ($F(1,35) = 10.485$, $p = 0.003$), as shown in Figure 8a. It should be noted that in the dynamic objective function experiment, there was no set interval at which operators were prompted to replan, but they were informed when the AS had a new proposed schedule that was at least 5% better than the current schedule.

- The length of time that operators spent replanning, where high performers spent less time evaluating new plans generated by the AS ($F(1,27) = 5.910$, $p = 0.022$) (Figure 8b).

- The rate of creating search tasks, where high performers created more search tasks throughout the mission ($F(1,35) = 18.697$, $p < 0.001$) (Figure 8c).

Finally, the workload level of the operators, as measured by utilization, was compared between the two performance cluster groups. While there was no significant effect for performance cluster ($F(1,35) = 2.472$, $p < 0.125$), there was a significant effect for time ($F(9,315) = 16.215$, $p < 0.001$), as utilization decreased throughout the mission (Figure 8d). This analysis showed that

62

high performers used the system as designed by replanning more frequently, spending less time evaluating new schedules generated by the AS, and creating more search tasks to encourage the UVs to explore new areas on the map. They were able to conduct more interventions to guide the automation without significantly increasing their workload, as measured by utilization.
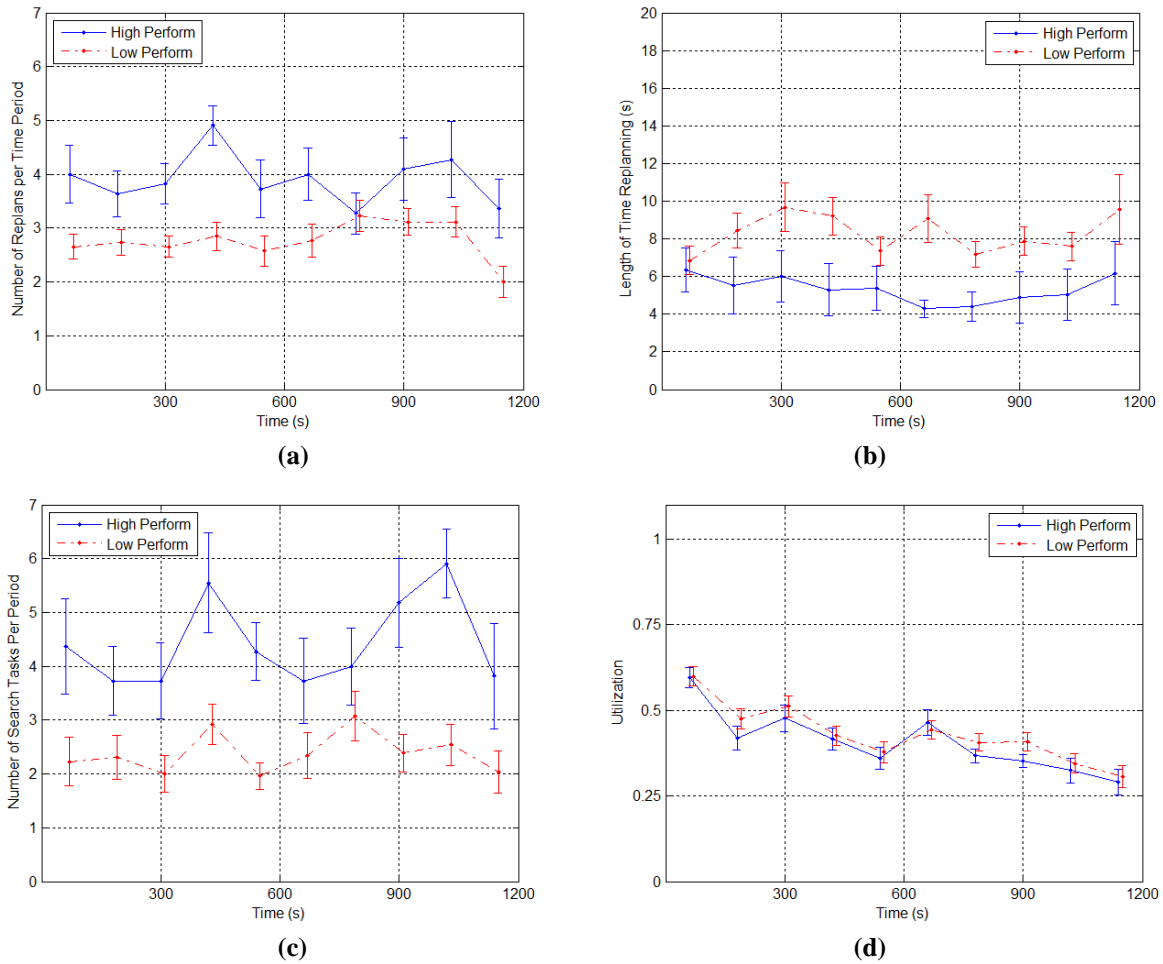
Figure 8. Differences in operator behavior between low and high performers. Standard Error bars are shown.

Once again, the hierarchical clustering analysis identified the number of meaningful groups in the data set, as shown in Figure 7a. While further data analysis could be conducted to determine what separates above average versus top performers, it is still instructive for model development that high performers intervened more frequently than low performers while making faster decisions. Why is it that high performers intervened more often to coach the automation? While data on operator trust is not explicitly available from this experiment, operators were asked to rate their satisfaction with the plans created by the AS after each mission on a Likert scale from

1-5 (low to high). This subjective rating can be used as a proxy variable for overall trust in the AS, as satisfaction of the operator's goals is a key component of the definition of trust (Lee & See, 2004). The High Performance cluster had a lower average rating of satisfaction than the Low Performance cluster, as shown in Figure 9a. This difference was significant according to a Mann-Whitney non-parametric test, $Z = -2.677$, $p = 0.007$. In order to further examine this trend, data on operator satisfaction with the AS from all 60 missions was analyzed. As the operator's satisfaction with the AS increased, overall area coverage performance decreased (Figure 9b). The figure only shows ratings between 1 and 4 because no operator rated their satisfaction level as 5 out of 5. The differences in performance across all 60 missions based on the operators' rating of their satisfaction with the AS are marginally significant according to a Kruskal-Wallis test, $\chi^2(3, N=59) = 7.262$, $p = 0.064$. Taking these ratings of satisfaction with the plans generated by the Automated Scheduler as a proxy measure of trust, it appears that higher performers had lower trust.
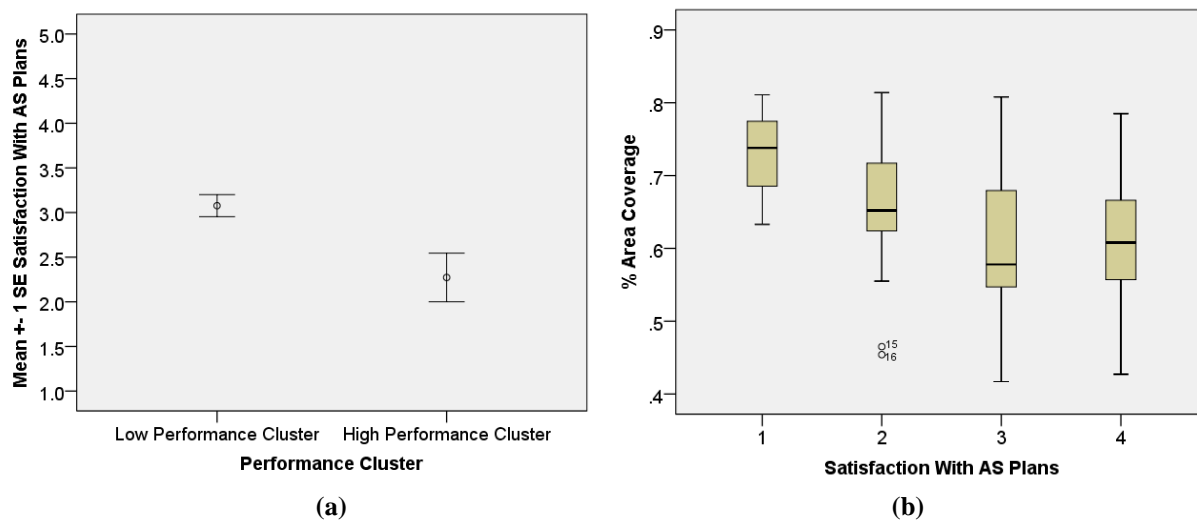
Figure 9. (a) Average subjective ratings of satisfaction with the plans created by the AS for each performance cluster. (b) Area coverage performance vs. operator ratings of satisfaction with AS plans (1=low, 4=high) for all missions.

Finally, a set of correlations were run among the variables, looking for relationships that should be captured in the model. While correlation does not necessarily imply causation, an understanding of how human operators use the OPS-USERS system can inform decisions to model correlated relationships as causation relationships. One of the most interesting relationships found was between the operator's satisfaction with the AS (a proxy measure of trust) and the number of search tasks that the operator creates per time interval ($\rho=-0.394$,

64

p=0.002). This indicates that operators who have lower trust in the AS may choose to intervene more frequently in order to guide the automation.

3.3 Dynamic Hypothesis

Based on this analysis, a dynamic hypothesis was developed that high performers adjusted to an appropriate level of trust in the AS earlier in the mission as compared to low performers. Previous research on the AS used in the OPS-USERS testbed has shown that the automated search process is suboptimal and can be improved either with a centralized global search algorithm (Whitten, 2010) or with a collaborative human assisting the AS (Cummings, et al., 2012), both of which extend the "effective planning horizon" of the search algorithm. Data analysis shows that higher performers understood the nature of these imperfections in the automation, reporting lower satisfaction/trust in the Automated Scheduler (p=0.007). They modified their behavior appropriately and used the system as designed by replanning more frequently (p=0.003), spending less time evaluating new schedules generated by the AS (p=0.022), and creating more search tasks (p<0.001) to encourage the UVs to explore new areas on the map.

Drawing an analogy to the anchoring and adjustment heuristic (Tversky & Kahneman, 1974), if operators can anchor to the correct expectation of AS performance earlier in the mission or adjust to the appropriate level of trust faster (through better feedback about the AS/system), performance should improve. The implementation of this dynamic hypothesis into a SD model is described in the next section.

3.4 CHAS Implementation

The CHAS model has been developed using SD modeling techniques, drawing from the results of the above data analysis, and supported where possible with examples from previously developed models and the human supervisory control literature. The model captures the six attributes that were identified in Chapter 2 as important to consider when modeling real-time human-automation collaborative scheduling: attention allocation and situation awareness, cognitive workload, trust in automation, human learning, automation characteristics, and human value-added through interventions.

CHAS is a computational model that can simulate the operations of a human-automation collaborative scheduling system throughout a hypothetical mission. It simulates the human operator, a team of UVs, and the automation at an abstract level, yet provides concrete metrics such as system performance, the frequency of certain operator decisions, and operator workload throughout the mission, not simply at the end or on average throughout the mission. This is especially helpful in dynamic missions where significant changes to the environment and the operator's behavior can occur. The model implements a set of equations which are calculated at discrete time steps using the Vensim® simulation software package.

A key fact to remember is that the model is meant to represent *goal-based* control of a team of UVs with the assistance of a decentralized planning algorithm (Clare & Cummings, 2011). The human operator only guides the high-level goals of the team of UVs (as opposed to guiding each individual vehicle), and collaborates with a decentralized planning algorithm, where each vehicle computes its locally best plan to accomplish the mission goals with shared information. This means that the human operator is monitoring the system and makes decisions to intervene throughout the mission in order to adjust the allocation of resources at a high level. In the next section, the model is described in further detail.

3.4.1 Model Overview

A simplified CHAS model is shown in Figure 10 which depicts the three major feedback loops: the Trust in Automation loop, the Expectations Adjustment loop, and the Cognitive Overload loop. These three loops consist of separate causal pathways through the model. A high-level discussion of the loops is presented below, while a more detailed discussion of the implementation of each loop is presented in Section 3.5.

The Trust in Automation loop, shown in the red dashed line box in Figure 10, draws from the "perception, cognition, action" loop in the human information processing model developed by Wickens and Hollands (2000). The loop represents how the operator's perception of the performance of the system impacts his or her trust in the automation and thus influences the operator's decisions to intervene in the operations of the semi-autonomous UVs. The operator has a time-delayed perception of how the system is performing. The operator's trust during the mission begins at an initial level and then adjusts based on the difference between the operator's

expectations of how well the system should be performing and the operator's perception of system performance. It is likely that the operator's trust has some inertia (Lee & Moray, 1994) and thus adjusts with a time delay. As the operator loses (or gains) trust in the AS, the operator will choose to intervene more (or less) frequently, for example by creating new tasks for the UVs or requesting a new schedule for the UVs. This decision to intervene has an impact on the operations of the team of semi-autonomous UVs, which influences the performance of the system, completing the feedback loop.
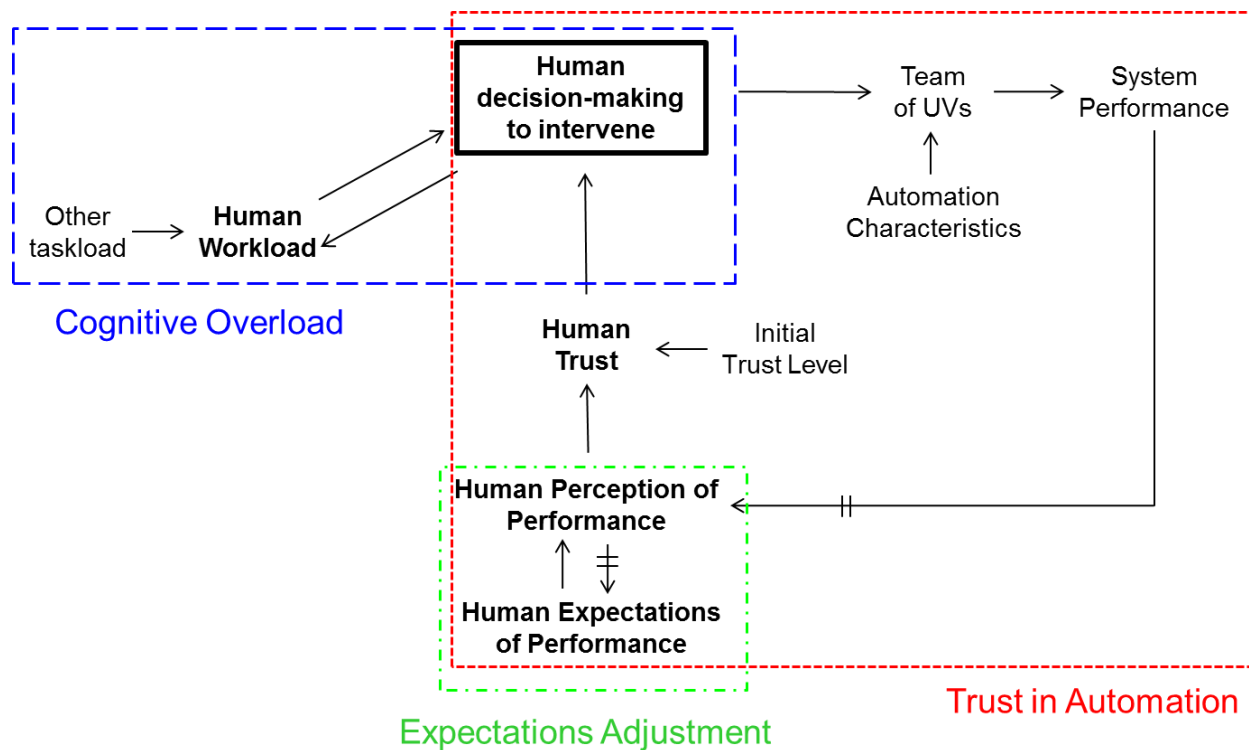
Figure 10. Simplified diagram of CHAS model.

Second, the Expectations Adjustment loop, shown in the green dashed line box, represents how the operator's expectations of performance can change throughout the mission. The operator's initial expectations of performance are likely set by training, previous experience, or by instructions from a supervisor. However, as the operator perceives how the system is actually performing, there is likely a time-delayed adjustment of the operator's expectations to conform to his or her perceived reality of how well the system is doing.

Third, the Cognitive Overload loop, shown in the blue dashed line box, represents the impact that excessive cognitive workload can have on system performance. The System Dynamics modeling

67

community typically separates the positive and negative effects of a variable into distinct loops (Sterman, 2000). The Trust in Automation loop, as previously described, captures the positive effects of increasing workload, assuming that an increasing rate of interventions leads to higher performance. It should be noted that this assumption does not hold for all systems, as previous studies have shown that frequent human intervention can potentially have a negative impact on automation (Beck, Dzindolet, & Pierce, 2005; Parasuraman & Riley, 1997), as some decentralized algorithms may need time to stabilize (Walker, et al., 2012). The automation in the OPS-USERS testbed has been found to be provably good, but suboptimal (Choi, et al., 2009; Whitten, 2010) and previous experiments have shown that a moderate rate of intervention results in higher performance than a low frequency of intervention (Clare, Maere, et al., 2012; Cummings, Clare, et al., 2010). Thus, it is assumed that operator interventions can improve performance.

The model captures the fact that the frequency with which the operator decides to intervene in the system also has an impact on human cognitive workload (Cummings, Clare, et al., 2010). The Cognitive Overload loop only captures the negative effects of *high* workload, and thus is dormant when the operator has low or moderate workload, having little effect on the model. Human workload is also driven by task load, i.e. the level of tasking that an operator is asked to perform by the system (Clare & Cummings, 2011). This model is specifically designed to simulate moderate to high task load missions and future research will investigate low task load, vigilance missions. The feedback loop is completed by modeling the potential for cognitive overload, where high levels of human workload can decrease the effectiveness of the operator's interventions in the system, thus decreasing system performance (Cummings & Guerlain, 2007; Nehme, 2009; Rouse, 1983; Schmidt, 1978).

It should be noted that the CHAS model underwent significant iteration. For example, "short-term" learning to use the graphical user interface more efficiently and analyze the information presented in the various displays more quickly was originally included in the model developed in this thesis, but was subsequently removed during the model reduction process (Appendix A). It was found that the impact of short-term learning on system performance and operator behavior was small. However, this model component could always be re-instated in the model if the

system being modeled had a significant short-term learning curve. The high-level differences between the original and final parsimonious CHAS model can be seen in Figure 11.
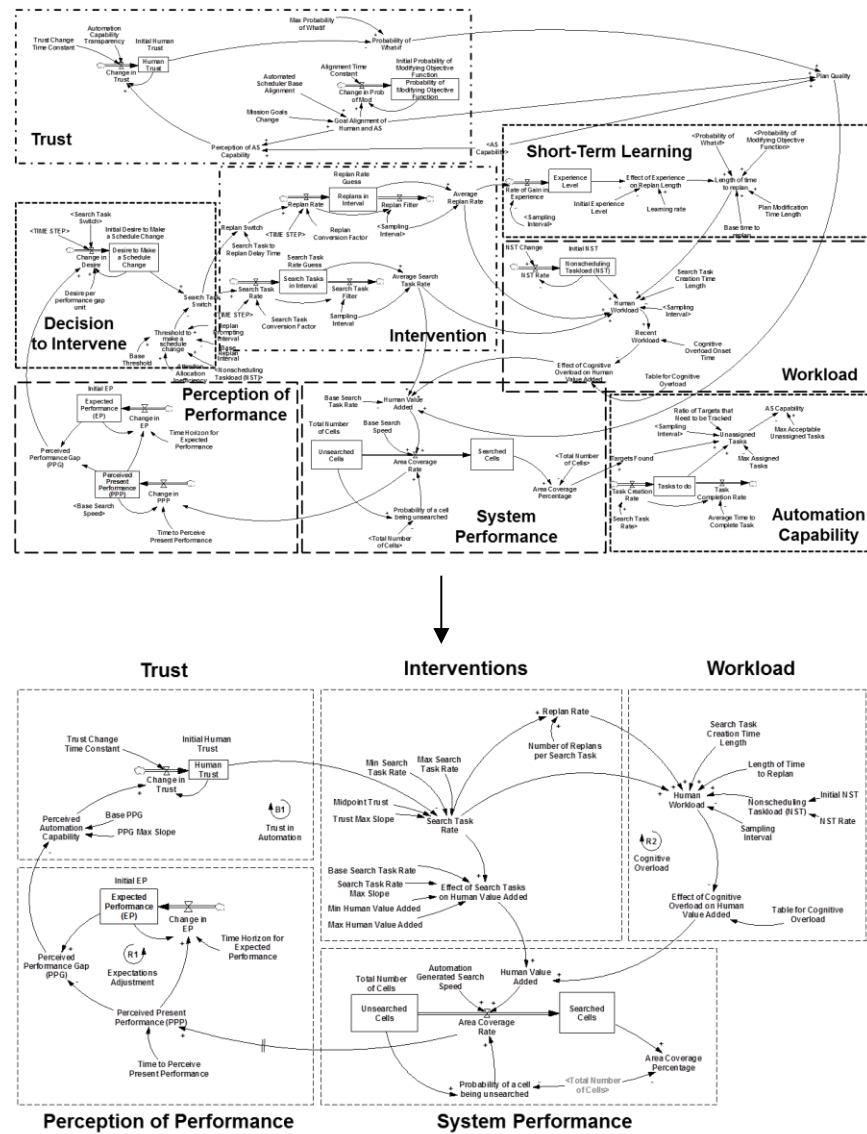
Figure 11. High-level diagram showing the original model reduced to the parsimonious model.

The full diagram of the reduced CHAS model is presented in Figure 12, showing the three main feedback loops. All of the equations and parameters that have been used in the final version of CHAS are listed in Appendix C. Many of the exogenous parameters can take distributions of values in Monte Carlo simulations to capture the impact of human variability, which is described in further detail in Chapter 4, along with testing of the model and fitting to experimental data. For ease of explanation, the three feedback loops are presented in five interconnecting modules, described in the following sections.
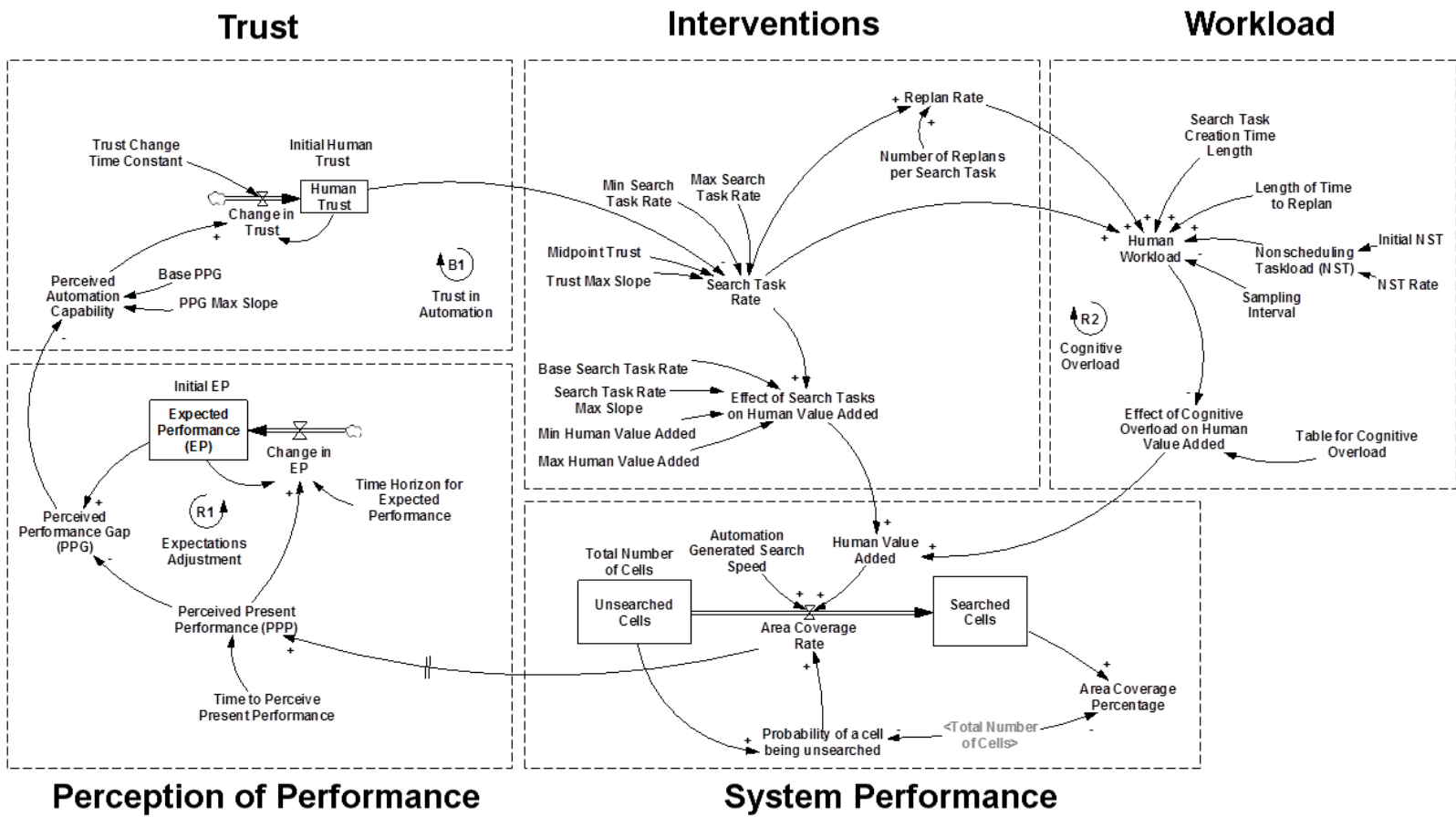
Figure 12. Collaborative Human-Automation Scheduling (CHAS) Model

3.4.2 System Performance Module

The first component of the CHAS model is the system performance module, shown in Figure 13. In order to properly model real-time human-automation collaborative scheduling, an effective, yet simple model of system performance is necessary. The system performance module aims to model one of the primary performance metrics from the OPS-USERS system, area coverage.
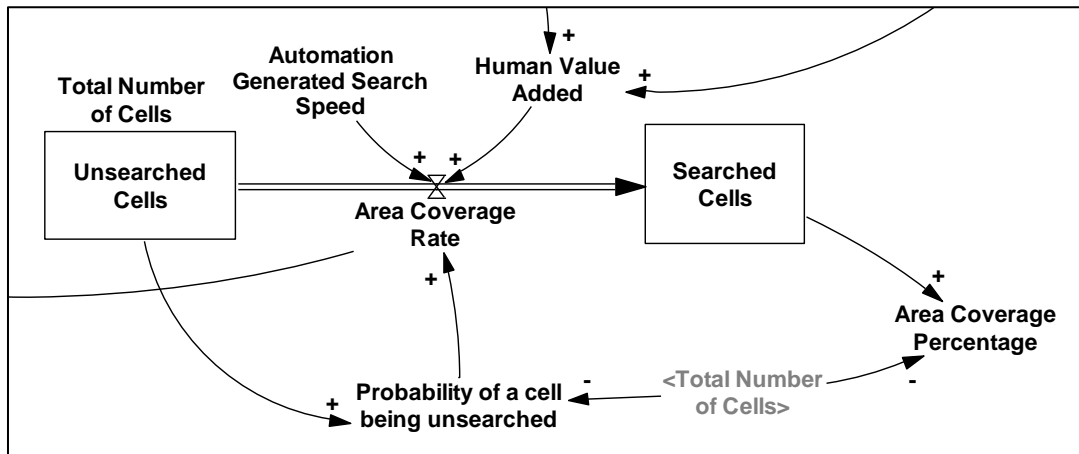
Figure 13. System performance module. "Total Number of Cells" is shown in gray to indicate that it is the same parameter, but used twice in the module.

The module is inspired by the diffusion SD model, which has been used to model the spread of new ideas, the adoption of new products, or the spread of contagious diseases (Sterman, 2000). In the OPS-USERS simulation, the total area to be searched by the team of UVs is discretized into 4,150 cells of equal area. Once a UV passes through that cell, the OPS-USERS testbed counts that cell as "searched" for the area coverage performance measure. The CHAS model represents the number of Unsearched Cells and the number of Searched Cells as "stocks." The Unsearched Cells stock is initialized to the Total Number of Cells. The "flow" between the two stocks is equal to the Area Coverage Rate.

While the area coverage performance metric is purely a count of cells which have been visited once, the actual search process in OPS-USERS is more complex than simply visiting every cell once. The system maintains a probability map which propagates the likelihood that a target is in each cell based on the last time that each cell was visited and a target motion model. The team of UVs re-visit previously searched cells based on this probability map (more details in Cummings, et al., 2012; Whitten, 2010). The CHAS model captures the impact of re-visiting previous cells

by decreasing the rate of covering new area over time through the Probability of a Cell Being Unsearched variable. The Probability of a Cell Being Unsearched is calculated by dividing the number of Unsearched Cells by the Total Number of Cells. Thus, the model assumes that as there are fewer unsearched cells, the area coverage rate will taper off.

The system performance module simplifies the complex human-automation collaboration for guiding a team of semi-autonomous UVs into two components: a) the Automation Generated Search Speed that the team of UVs would have without any human input and b) the Human Value Added to the system. As discussed in Chapter 2, in a *goal-based* architecture (Clare & Cummings, 2011) the vehicles are semi-autonomous and with the guidance of the AS, can conduct much of the mission on their own. The human operator only guides the high-level goals of the vehicles, as opposed to guiding each individual vehicle. Thus, the CHAS model captures the contribution of the automation through an exogenous parameter, Automation Generated Search Speed, which is dependent on the number of UVs, speed of the UVs, capabilities and sensors of the UVs, and the algorithm that the AS employs.

The human contribution to the collaboration is represented in the model as an endogenously-calculated Human Value Added variable. This enables the model to separately capture the contribution of the automation and the human to area coverage performance. While the operator cannot increase the search speed of the UVs, the operator can indirectly adjust the search patterns of the UVs, which can increase (or decrease) the rate of searching *new* cells. Previous research on the AS used in the OPS-USERS testbed has shown that the automated search process is suboptimal and can be improved either with a centralized global search algorithm (Whitten, 2010) or with a collaborative human assisting the AS (Cummings, et al., 2012), both of which extend the "effective planning horizon" of the search algorithm.

While the calculation of Human Value Added itself is non-linear (Sections 3.4.5 and 3.5), it was decided to model the contribution of Human Value Added to the Area Coverage Rate as an *additive* factor in order to capture two important facts about real-time human-automation collaborative scheduling: a) the system could potentially conduct some of the mission without any meaningful contribution from the human operator and b) while human operators can increase the performance of these collaborative systems, it is also possible for them to hurt the

72

performance of a system either intentionally or unintentionally. Thus, the Human Value Added variable is calculated with units of cells/second, the same units as Automation Generated Search Speed and Area Coverage Rate, to capture how much the operator is increasing or decreasing the rate of searching *new* cells.

The Area Coverage Rate is calculated using Equation 1, where the sum of Automation Generated Search Speed and Human Value Added is multiplied by the Probability of a Cell Being Unsearched. This represents the fact that once there are few unsearched cells remaining, the rate of area coverage must approach zero. Area Coverage Percentage, the primary system performance metric for this model, is calculated by dividing the number of Searched Cells by the Total Number of Cells parameter.

$$
\begin{aligned}
Area\ Coverage\ Rate \\
= Probability\ of\ a\ Cell\ Being\ Unsearched\ * \\
(Automation\ Generated\ Search\ Speed + Human\ Value\ Added)
\end{aligned} \tag{1}
$$

There are three major simplifying assumptions that the system performance module makes. First, the model assumes that the total number of cells possible to search can be quantified and measured. This amount is known in the OPS-USERS simulation and is shown as the total area on the Map Display (Figure 4). Second, the model assumes that Automation Generated Search Speed can be captured in a single exogenous parameter. Third, the model assumes that the human contribution to the collaboration can be separately calculated and summed with Automation Generated Search Speed to produce a measure of area coverage rate.

To test these simplifying assumptions about the complex human-automation collaboration, it was important to measure how the team of UVs under purely automation control would perform without any contribution from the human operator with regards to the guidance of the UVs. Thus, for this test, an "obedient" human operator was used, as has been used in previous experiments to make a similar comparison (Cummings, et al., 2012). An obedient operator always agrees with the AS proposed schedule and never conducts any interventions (such as manually modifying the schedule, or creating new search tasks), in effect always trusting that the AS was correct in its guidance of the UVs. The human operator still had an important role to play with regards to identifying targets, integrating information from the command center received through the chat box, and approving the destruction of hostile targets.

Using the same testbed, mission length, and experimental conditions described in Section 3.2, an obedient human mission was conducted and the results are shown in Figure 14. The actual data collected from the testbed are shown in the red squares. First, the data shows a saturation-type curve, indicating that the *rate* of covering new area slows down over time, supporting the structure of the system performance module. Second, the percent area covered in the "obedient human" condition was 56.2%. In comparison, the average human operator in the experiment described in Section 3.2 achieved area coverage of 61.7%, a 10% increase in performance due to human value added (blue diamonds in Figure 14). The high performer group in the experiment achieved average area coverage of 76.6%, a 36% increase in performance over the "obedient human" condition (green triangles in Figure 14). These results support the assumption that the human operator is adding value over the automation generated s.

The system performance module was used to simulate this test, with the Human Value Added set to zero, essentially turning off all other modules in the CHAS model. The model simulation is shown in the blue line in Figure 14. The model was able to achieve a good fit to this data, with an R^2 value of 0.9932 and a RMSE of 0.0143. This test also enabled the estimation of the Automation Generated Search Speed parameter, set to 2.9 cells/second.
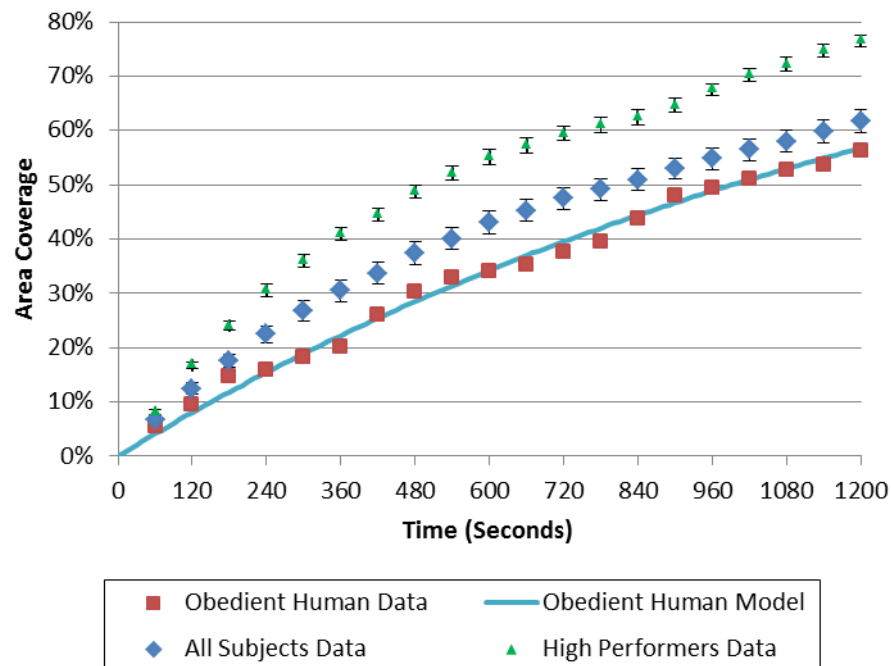
Figure 14. Obedient human area coverage performance: model vs. experimental data. Aggregate and high performer data from previous experiment shown for comparison. Standard error bars shown.

The ability of the module to adequately capture the key drivers of system performance and accurately calculate performance over time is explored in greater detail in Chapter 4.

3.4.3 Perception of Performance Module

One of the major attributes identified in Chapter 2 that should be captured in a model of human-automation collaborative scheduling is attention allocation and Situation Awareness (SA). The Perception of Performance module captures both attention allocation and the three levels of SA (Endsley, 1995).

There are two major components to the module, as shown in Figure 15. First, there is the Perceived Present Performance (PPP), which takes as an input Area Coverage Rate. PPP is a measure of performance, delayed by the parameter Time to Perceive Present Performance (TPPP), which is an assumption about how long it takes the operator to detect changes in the area coverage rate. This delay is implemented as a third-order exponential smooth, which has been used in previous SD models to represent human perception, decision-making, and response delays (Naill, 1973; Senge, 1980; Sterman, 2000). This method captures two facets of human perception: a) operators do not immediately perceive changes in system performance, this belief changes only after some time has passed and b) operators may filter out high-frequency, short-term changes in system performance and this process can be represented mathematically through averaging/smoothing.

TPPP is also an implicit measure of attention allocation efficiency, which was described in Chapter 2 as an important attribute to capture in a model of human-automation collaborative scheduling. Operators engaged in these dynamic, high workload environments must both concentrate attention on the primary task of monitoring UV progress and system performance while also being prepared for various alerts, such automation notifications about potential changes to the vehicle schedules. The allocation of attention between these two can incur cognitive switching costs that negatively impact overall system performance (Miyata & Norman, 1986). Poor attention allocation has been shown to be a significant contributor to poor operator performance in single operator control of multiple unmanned vehicles (Crandall & Cummings, 2007; Goodrich, et al., 2005). Thus an operator with higher attention allocation who is

better able to handle the rapid task switching required would have a lower TPPP and detect changes in system performance faster than an operator who struggles with multitasking.
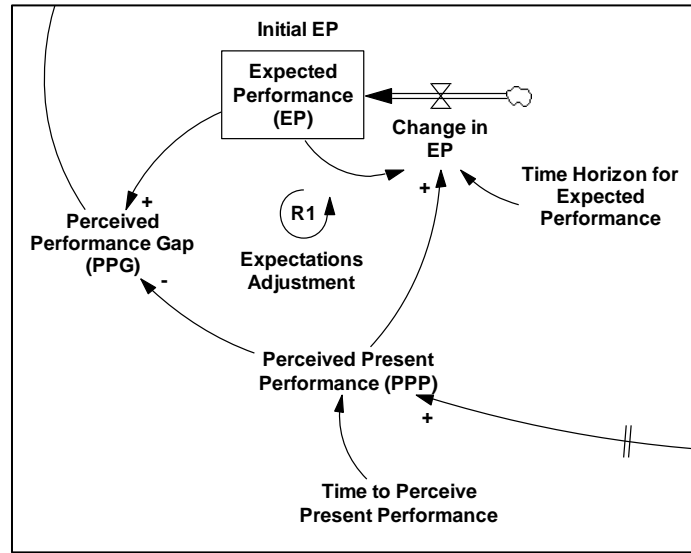
Figure 15. Perception of performance module.

Second, the CHAS model assumes that the operator begins the mission with some expectation of how well the system should be performing, Expected Performance (EP). The Initial EP may be based on many factors, such as prior experience or training with the system, or various demographic characteristics. The operator's expectations of how well the system should be performing, however, are not static. The CHAS model represents the EP as a "floating anchor" which adjusts via first-order exponential smoothing to the perceived performance of the system over time. This formulation was inspired by the TREND function (Sterman, 1987a; Sterman, 2000), which is based on a model of how humans forecast future conditions, such as inflation or energy consumption, based on previous data. The Time Horizon for Expected Performance (THEP) is an estimate of the adjustment lag required for operators to change their expectations, similar to the estimate of the time horizon for the forecasting process (Sterman, 1987a; Sterman, 2000).

Finally, the percent difference between PPP and EP is called the Perceived Performance Gap (PPG), calculated via Equation 2. When the operator perceives that system is not performing up to his or her expectations, there is a positive PPG, and vice versa.

$$PPG = \frac{EP - PPP}{EP} \tag{2}$$

76

One of the advantages of this formulation of the Perception of Performance module is that it explicitly represents the three levels of SA (Endsley, 1995) and how they can change over time throughout a mission:

- Perception (Level I): PPP represents the operator's delayed perception of performance.

- Comprehension (Level II): PPG represents the operator's understanding of the situation, specifically the gap between expected and actual performance.

- Projection (Level III): EP, as a floating anchor of the operator's expectation of performance, represents the operator's projection of future performance.

This module makes two simplifying assumptions. The model assumes perfect operator perception of the area coverage rate, with only a time delay. However, humans are not perfect sensors of information, especially when there is no explicit indication of the system performance. Second, as mentioned above, the model assumes that the operator begins the mission with a certain expectation level of performance, but adjusts these expectations throughout the mission. Both of these assumptions are evaluated in Chapters 4 and 5.

3.4.4 Trust Module

Chapter 2 described the importance of trust in a human-automation collaborative scheduling system. The Trust Module draws from a previous computational model of human trust in automation. Gao and Lee (2006) modeled human trust in a supervisory control setting as dependent on the operator's perception of the capability of the automation. They define automation capability as "the reliability of the automation in terms of fault occurrence and general ability to accomplish the task under normal conditions" (Gao & Lee, 2006, p. 946). In their model, the operator's trust began at an initial level, given by a parameter, and adjusted with some inertia, meaning that some operators adjusted their trust faster than others. Gao and Lee represented the inertia of trust through a trust time constant that governed how quickly trust could change. A larger trust time constant meant that the operator put greater weight on new information that was perceived in terms of the capability of the automation and thus had lower inertia of trust. Finally, the operator's decisions about the next action to take were influenced by the operator's trust in the automation. Portions of the Gao and Lee (2006) model were adapted

for use in the CHAS model to capture the impact of trust on human-automation collaboration for control of multiple UVs.

The trust module, as shown in Figure 16, begins by calculating the Perceived Automation Capability based on the Perceived Performance Gap (PPG). Perceived Automation Capability, which is expressed as a percentage between 0-100%, then drives changes in Human Trust. The model assumes that human operators adjust their trust in the automation based not upon their absolute perception of the performance of the system, but on the relative difference between what they perceive and how well they expect the system to be doing. For example, an operator with very low expectations of performance could have higher trust in the automation based on their perception of how well the system is doing. This is supported in the human supervisory control literature, for example, Lee and See (2004, p. 53) explained that trust in automation "concerns an expectancy or an attitude regarding the likelihood of favorable responses," but that trust changes based on the operator's perception of the current capability of the automation as compared to their expectation of capability. Also, Lee and Gao (2006) used the occurrence of automation faults, i.e. performance that was lower than expected, to model the capability of the automation as a measure of reliability.
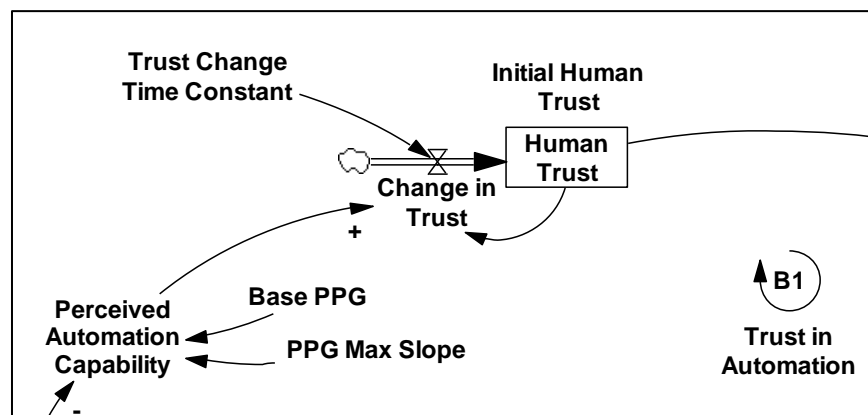
Figure 16. Trust module.

The model assumes that increases in PPG should cause a decrease in Perceived Automation Capability and vice versa. It was decided to model the relationship between PPG and Perceived Automation Capability with an inverse logit function, as shown in Figure 17. This non-linear relationship is supported by previous research, as Lee and See (2004, p. 72) wrote that "trust is a nonlinear function of automation performance and the dynamic interaction between the operator

78

and the automation…some evidence suggests that below a certain level of reliability, trust declines quite rapidly." Thus, in the CHAS model, below a certain level of reliability (above a certain level of PPG), trust (driven in the CHAS model by Perceived Automation Capability) declines rapidly. A logit function also maintains the Perceived Automation Capability between 0-100% even at extremely high or low values of PPG.

This relationship is captured by the logit function shown in Figure 17, which shows a fairly rapid decline in Perceived Automation Capability as the operator moves from a 0% PPG to a 30% PPG. The shape of this curve is dependent on the characteristics of the system being modeled and can be varied in the CHAS model by two parameters as seen in Equation 3: Base PPG and PPG Max Slope. The Base PPG determines the point at which Perceived Automation Capability crosses 50%. PPG Max Slope determines the maximum slope of the curve at the midpoint. Equation 3 shows the calculation for Perceived Automation Capability. By capturing this relationship through these two parameters, it enables sensitivity analysis of how the model outputs change with different curves, which is explored further in Section 4.3.

$$Perceived\ Automation\ Capability = (1 - 1/(1 + e^{-4*PPG\ Max\ Slope*(PPG-Base\ PPG)}) \quad (3)$$
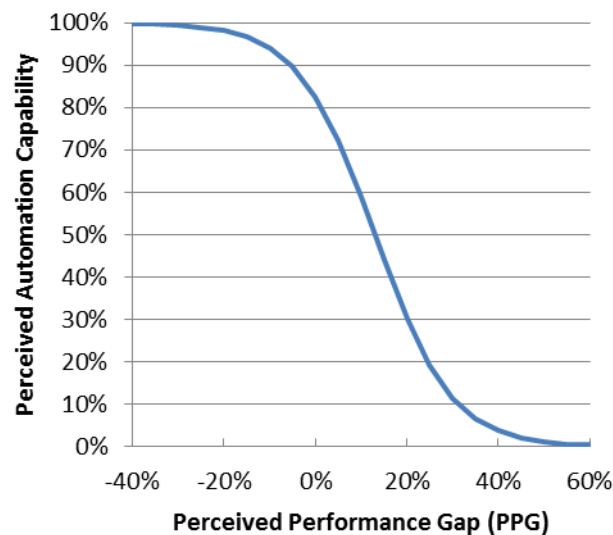
Figure 17. Notional relationship between Perceived Performance Gap (percent difference between expected and perceived performance) and Perceived Automation Capability.

Next, the model takes as an input parameter the Initial Human Trust in the AS, which can vary widely based on the operator's prior knowledge, past experiences, and training (Lee & Moray,

1994; Moray, et al., 2000). Both the Initial Human Trust and the Human Trust variable are expressed as a percentage between 0-100%, matching the definition of Perceived Automation Capability.

Trust is often dynamic and can fluctuate throughout a mission based on the operator's perception of how well the automation is performing (Lee & Moray, 1992; Muir & Moray, 1996). A number of studies have found that human trust has inertia, where automation errors do not necessarily cause instantaneous loss in trust, but recovery in trust from severe failures can also be slow (Lee & Moray, 1994; Lewandowsky, et al., 2000; Parasuraman, 1993). To reflect the dynamic nature of trust, the model adjusts the operator's trust via first-order smoothing to the Perceived Automation Capability with a time delay. Just as Lee and Gao (2006) used in their model, the time delay is determined by an exogenous parameter, the Trust Time Change Constant, which is representative of the operator's trust inertia. Both the Initial Human Trust and Trust Time Change Constant are estimated through model fitting to experimental data, as described in Chapter 4.

Once again, there are a number of simplifying assumptions in this module. First, the model assumes that human trust in the AS is negatively dependent on the PPG. Second, the model assumes that the non-linear relationship between PPG and trust can be captured through a logit function. Third, the model assumes that trust begins at a given level, potentially different for each operator, and adjusts with some inertia. All of these assumptions will be evaluated in Chapters 4 and 5.

3.4.5 Interventions Module

The interventions module is shown in Figure 18. Two types of interventions were specified in the CHAS model. First, the operator could create new search tasks, which enabled the human operator to guide the automation and prevent myopic behaviors by encouraging the UVs to search in unsearched areas the human felt were likely to contain new targets. Second, the operator could "replan" by asking the AS to generate a new schedule for the team of UVs. This new schedule must be approved by the operator before the team of UVs will enact the new schedule. These two interventions were chosen because the data analysis discussed previously in Section 3.2 revealed that there were significant differences between low and high performers in

these two types of interventions. Other interventions to modify the automation-generated schedules directly, such as "what-if assignments" to attempt to manually force a single task into the schedule and modifying the objective function of the AS (Section 3.2) were removed from the CHAS model in the model reduction process described in Appendix A.
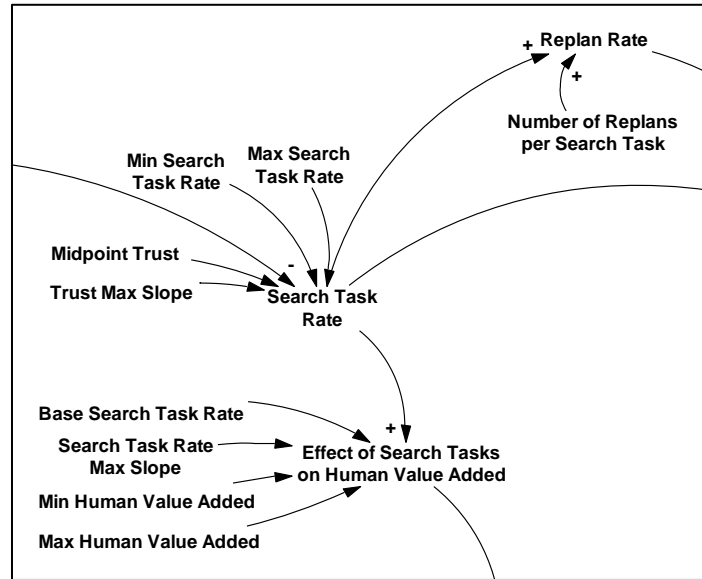
Figure 18. Interventions Module.

First, the module assumes that the rate of intervening by creating new search tasks is negatively dependent on Human Trust level (higher trust, less likely to intervene). As was explained in Section 3.2, data from a previous experiment showed that the operator's subjective rating of satisfaction with the AS (a proxy variable for trust) was negatively correlated with the rate of creating new search tasks, $\rho=-0.394$, $p=0.002$. It was decided to model the relationship between Human Trust and Search Task Rate with an inverse logit function, as shown in Figure 19. A non-linear logit relationship between trust and interventions is supported by previous empirical results (Lee & Moray, 1994; See, 2002). In particular, Lee and Gao (2006) modeled the relationship between trust and the likelihood that an operator would use automatic versus manual control using a logit function.

Empirical evidence supporting the shape of this curve is also presented in Figure 19. The data shown in the plot is from the previous experiment described in Section 3.2. While explicit data on operator trust is not available from this experiment, operators were asked to rate their satisfaction with the plans created by the AS after each mission on a Likert scale from 1-5 (low

to high). This subjective rating can be used as a proxy variable for trust in the AS. For the purposes of this plot, the ratings of satisfaction with the plans created by the AS were converted to the trust scale (0-100%) simply by dividing by 5. The yellow diamond shows the average for all missions in terms of the proxy trust measure and search task rate. Data from the high and low performance clusters identified in Section 3.2 are shown in green triangles. The blue squares show the average search task rate for groups of operators based on their rating of satisfaction with the AS. There are four groups (ratings 1-4), as no operator rated their satisfaction level as 5 out of 5. It is apparent from the data that the relationship is non-linear, as the decline in search task rate with increasing trust begins sharply, but flattens out at higher trust levels.
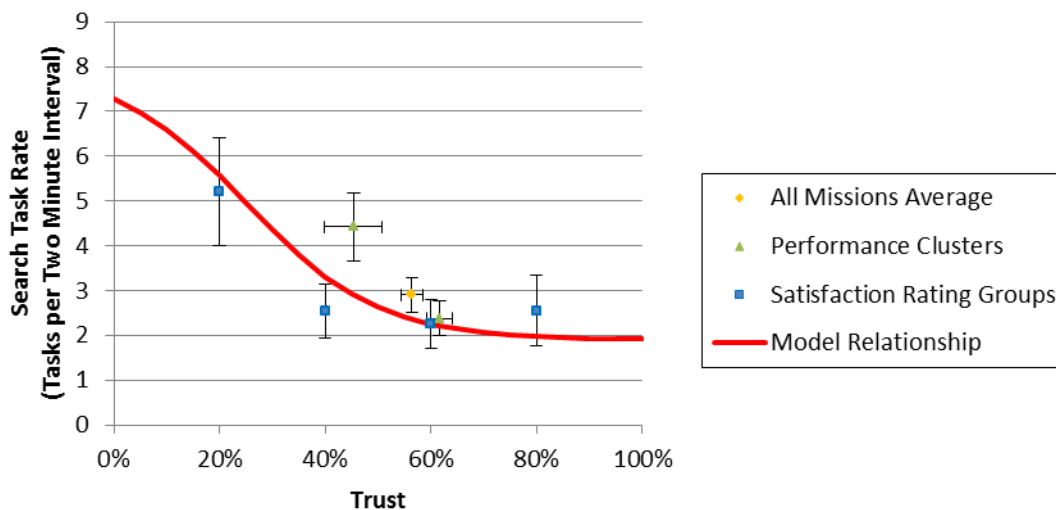
Figure 19. Relationship between Human Trust and Search Task Rate. Empirical data shown with ±1 Standard Error bars.

The shape of this curve can be varied in the CHAS model of the OPS-USERS environment by four parameters as seen in Equation 4: Max Search Task Rate, Min Search Task Rate, Midpoint Trust, and Trust Max Slope. The Max and Min Search Task Rate set the minimum and maximum values for the curve. The Midpoint Trust determines the midpoint of the logit curve, i.e. the trust level at which the curve crosses halfway between the Max and Min Search Task Rate. Trust Max Slope determines the maximum slope of the curve at the midpoint. Equation 4 shows the calculation for Search Task Rate. By capturing this relationship through these three parameters, it enables sensitivity analysis of how the model outputs change with different curves, which is explored further in Section 4.3.

$$Search\ Task\ Rate$$
$$= (Max\ Search\ Task\ Rate - Min\ Search\ Task\ Rate) *$$
$$\left(1 - \frac{1}{1+e^{-4*Trust\ Max\ Slope*(Human\ Trust-Midpoint\ Trust)}}\right) + Min\ Search\ Task\ Rate \quad (4)$$

Next, the interventions module captures the Replan Rate through a direct relationship with the Search Task Rate, as shown in Equation 5. The Number of Replans per Search Task parameter scales the rate of creating search tasks to the rate of replanning. The relationship between search task creation and replanning is a result of the design of the testbed. There is a direct system need, once a search task has been created, to replan in order to assign that new task to the team of UVs. The OPS-USERS system actually displays a notification every time a search task is created, encouraging the operator to replan to assign the new task. Data from the previous experiment described in Section 3.2 supports this relationship, as the search task rate was significantly correlated with the replan rate, $\rho=0.567$, $p<0.001$. While the search task rate and replan rate are highly correlated, the CHAS model only represents the impact of search task creation on the human value added to system performance, as described below. It is still important to calculate the replan rate to enable an accurate estimation of workload, as described in Section 3.4.6.

$$Replan\ Rate = Search\ Task\ Rate * Number\ of\ Replans\ per\ Search\ Task \quad (5)$$

Finally, the interventions module calculates the Effect of Search Tasks on Human Value Added. Data from the previous experiment described in Section 3.2 showed that the rate of creating new search tasks (an intervention meant to guide the search process of the team of UVs) was significantly correlated with area coverage performance, $\rho=0.446$, $p<0.001$. This supports the assumption that there is a positive relationship between the rate of creating search tasks and the human value added to system performance.

In order to estimate the relationship between Search Task Rate and Human Value Added, the simple test model shown in Figure 20 was used. This model is identical to the system performance module presented previously except that instead of separating the Automation Generated Search Speed and Human Value Added, it uses a single Search Speed parameter to drive the model. Also, there are no feedback loops linking the area coverage rate to operator trust, interventions, or workload.
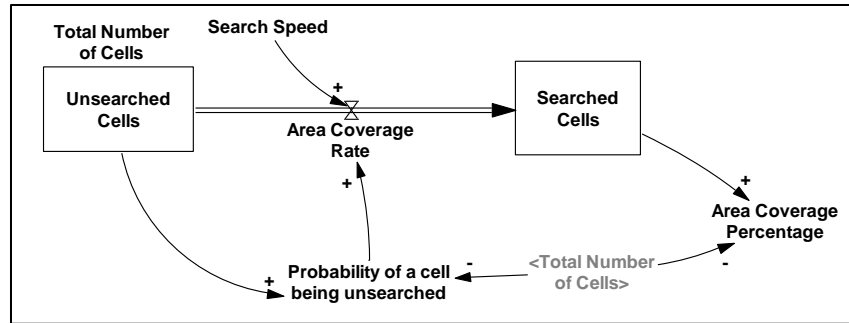
Figure 20. Test model for estimating relationship between Search Task Rate and Human Value Added.

The Search Speed input parameter was varied in order to fit the test model to the area coverage data from three groups in the experimental data set described in Section 3.2: low performers, high performers, and aggregate data for all missions. The results are shown in Table 1. Given that the *Automation Generated* Search speed was estimated in Section 3.4.2 to be 2.9 cells/second, the Human Value Added could be estimated by subtracting 2.9 from the fitted Search Speed value of each group, as shown in the fourth column of Table 1. The results show that low performers were actually hurting the performance of the system (either intentionally or unintentionally) as compared to how the system would have performed under automation-only control. The average performer added some value to system performance, while high performers added the most value, contributing 68% of the value that the automation was contributing. As the data analysis in Section 3.2 showed, it is likely that operators increased the rate of covering new area by creating more search tasks to encourage the UVs to explore new areas on the map.

Table 1. Estimated relationship between Average Search Task Rate and Human Value Added.

Data Set	Average Search Tasks Created Per Two Minutes	Fitted Search Speed	Human Value Added
Low Performers	2.38	2.4	-0.5
All Missions	2.90	3.6	0.7
High Performers	4.42	4.9	2

By plotting Average Search Task Rate and Human Value Added, as shown in Figure 21, a non-linear logit relationship was estimated. The CHAS model assumes that operators who create fewer than 2.5 search tasks per two minute interval are actually lowering the performance of the system as compared to the Automation Generated Search Speed. The model also assumes that the contribution of the human operator to the system is limited in both the positive and negative direction, with the minimum Human Value Added set to -1 cells/second and the maximum

Human Value Added set to 2 cells/second. As a model construct, these limits ensure that the model behaves appropriately at extreme conditions. Without these limits, for example, the Effect of Search Tasks on Human Value Added could scale linearly down to the point at which the Human Value Added cancels out the Automation Generated Performance, causing the Area Coverage Rate to become negative, which is physically impossible. The non-linear logit relationship enables this appropriate model behavior at extreme conditions and further extreme conditions testing is described in Chapter 4.
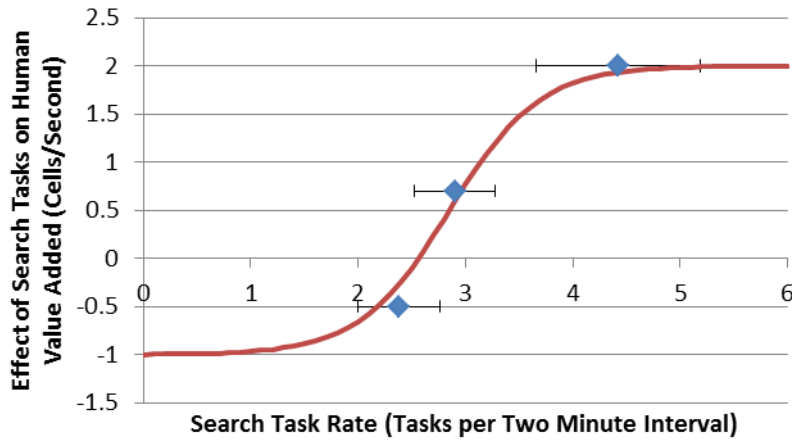
**Figure 21. Relationship between Search Task Rate and Effect of Search Tasks on Human Value Added.
Empirical data shown with ±1 Standard Error bars.**

The shape of this curve can be varied in the CHAS model by four parameters as seen in Equation 6: Max Human Value Added, Min Human Value Added, Base Search Task Rate, and Search Task Rate Max Slope. The Max and Min Human Value Added set the minimum and maximum values for the curve. The Base Search Task Rate determines the search task rate at which the curve crosses halfway between the Max and Min Human Value Added. Search Task Rate Max Slope determines the maximum slope of the curve at the midpoint. Equation 6 shows the calculation for Effect of Search Tasks on Human Value Added. This effect is combined with the impact of cognitive overload to calculate Human Value Added (Section 3.5). By capturing this relationship through these four parameters, it enables sensitivity analysis of how the model outputs change with different curves, which is explored further in Section 4.3.

$$Effect\ of\ Search\ Tasks\ on\ Human\ Value\ Added$$
$$= \left(\frac{Max\ Human\ Value\ Added - Min\ Human\ Value\ Added}{1 + e^{-4 * Search\ Task\ Rate\ Max\ Slope * (Search\ Task\ Rate - Base\ Search\ Task\ Rate)}} \right)$$
$$+ Min\ Human\ Value\ Added \tag{6}$$

To summarize, this module contains a number of assumptions. First, the model assumes that the rate of creating new search tasks is negatively dependent on human trust. Second, the model assumes that the non-linear relationship between trust and search task rate can be captured through a logit function. Third, the model assumes that the rate of replanning is directly dependent on the rate of creating search tasks. Fourth, the model assumes that the Human Value Added to system performance is positively dependent on the rate of creating search tasks. Finally, the model assumes that the non-linear relationship between search task rate and Human Value Added can be captured through a logit function. While empirical data was presented throughout this section to support these assumptions, further evaluation of these assumptions will be presented in Chapters 4 and 5.

3.4.6 Workload Module

As described in Chapter 2, it has been shown in several previous studies that human cognitive workload has a significant impact on both human and system performance (Clare & Cummings, 2011; Cummings, Clare, et al., 2010; Cummings & Nehme, 2010). As theorized in the Yerkes-Dodson curve (1908), up to a certain point, increased workload can be beneficial to performance. Once the operator reaches what is referred to as cognitive overload, performance begins to suffer (Figure 22). Miller (1978) surveyed a large number of studies on the ability of individuals to process information and found a robust inverted U-shaped relationship between the rate of information inputs (a proxy for workload) and the ability of individuals to produce correct responses (performance).
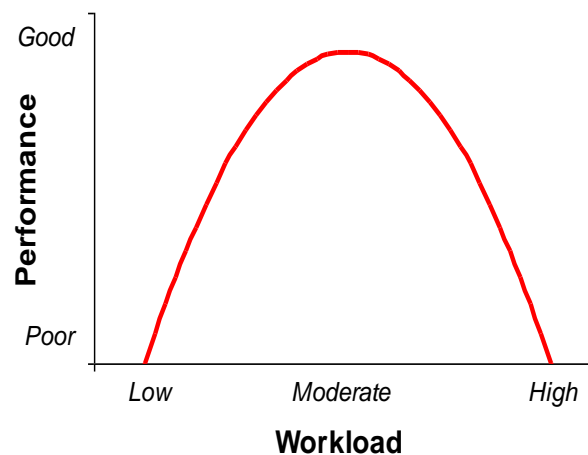
Figure 22. Notional diagram of the Yerkes-Dodson curve.

The System Dynamics modeling community avoids implementing U-shaped non-linear functions for two reasons: a) to ensure that all causal links in the model have unambiguous polarity and b) a U-shaped relationship indicates the presence of multiple causal pathways between the input and output (Rudolph & Repenning, 2002; Sterman, 2000). Thus, the increasing and decreasing effects of the Yerkes-Dodson curve are separated in the CHAS model. While this may add complexity to the model, it also clarifies which side of the Yerkes-Dodson curve the model is operating on at all times. The Trust in Automation loop, as previously described, captures the positive effects of increasing workload, where an increased rate of creating search tasks leads to higher performance (as shown in Section 4.2.1). The workload module specifically captures the negative effects of cognitive overload in real-time human-automation collaborative scheduling.

The workload module is shown in Figure 23. Human Workload is measured through a utilization metric, calculating the ratio of the total operator "busy time" to the total mission time. This captures the percentage of time that the operator is engaged in a goal-directed task, not monitoring the system. Operator "busyness" can serve as a useful proxy measure of mental workload (Wickens & Hollands, 2000).
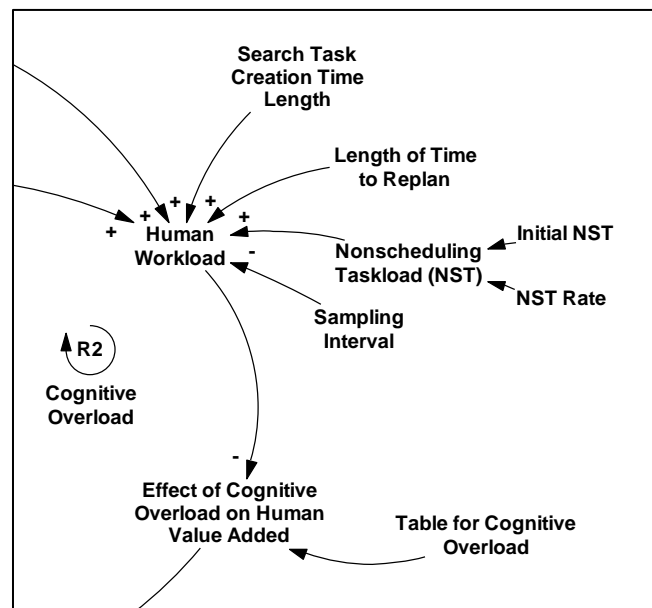
Figure 23. Workload module.

A key assumption in the CHAS model is that operators must deal with non-scheduling activities during the mission in addition to the two scheduling tasks of creating search tasks and replanning

(evaluating new schedules created by the AS). For the purposes of this thesis, these two types of activities are called scheduling activities and Nonscheduling Task Load (NST). Scheduling activities have two defining characteristics: a) the operator decides when to conduct these activities and b) the activity is directly related to scheduling the UVs. In contrast, NST is defined as the level of tasking that an operator is asked to perform, excluding scheduling activities. There were three activities that fell in this category in the OPS-USERS testbed: visually identifying or re-designating targets, approving weapons launch, and reading and answering chat messages. For all three of these activities, the operator was prompted to do the activity, either through a pop-up window or an auditory alert (see Appendix D for more details).

The CHAS model endogenously calculates the rate of creating search tasks and the rate of replanning, as the operator decides when to conduct these activities. The model assumes that the required utilization generated by NST can be captured exogenously based on data gathered from the OPS-USERS testbed. The required utilization due to NST was calculated from aggregate experimental data (Section 3.2.2), using two-minute intervals, as shown in red in Figure 24. Self-imposed utilization from scheduling activities (creating search tasks and replanning) is shown in blue stripes.
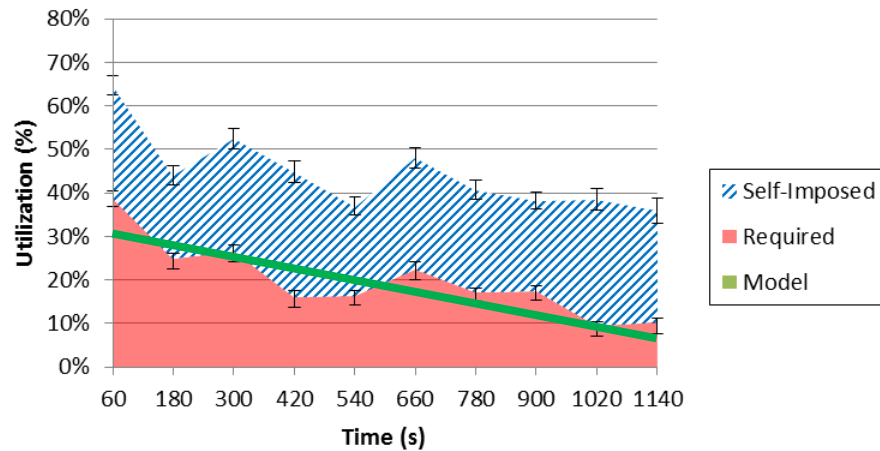
Figure 24. Utilization due to self-imposed scheduling activities and required utilization due to Nonscheduling Task load (NST). Standard error bars are shown.

A number of observations can be made from this data. First, the required utilization due to NST appears to decline over time. A repeated measures ANOVA of the required utilization due to NST indicates a significant effect for time, $F_{(9,531)} = 55.140$, $p < 0.001$. Descriptive statistics

for this ANOVA analysis are presented in Appendix D. Second, it appears that this decline is roughly linear. The green line shown in Figure 24 captures most of the variance in required utilization due to NST, with an R^2 value of 0.72.

Why is NST declining over time? The main reason for this linear decline in NST is that most of the visual target identification activities and chat messages occur earlier in the mission and are less frequent towards the end of the mission. This linear decline appears to hold except for a small uptick in NST in the interval following 600 seconds into the mission. This increase in NST is clearly linked to a ROE change allowing the destruction of hostile targets, which occurs at 600 seconds. A similar analysis of required utilization due to NST for different OPS-USERS experiments also showed a roughly linear decline in NST (Appendix D).

Overall, the assumption of an exogenously defined, linearly declining NST seems to be appropriate for modeling the OPS-USERS testbed under the experimental conditions described in Section 3.2 (see Section 4.2.2 for an analysis of NST for a different OPS-USERS experiment with higher task load). For the experimental conditions in Section 3.2, the CHAS model representation of NST follows the green "model" line shown in Figure 24, with the required utilization due to NST starting at 30% followed by a linear decline. The CHAS model calculates the Nonscheduling Task load (NST) from two exogenous parameters: Initial NST and NST Rate, as shown in Figure 23. NST begins at a level set by Initial NST and either linearly increases, linearly decreases, or remains the same depending on the NST Rate of change parameter.

To conclude the calculation of human workload, it should be noted that the experimental data set used to build this model measured the rate of interventions (search task creation, replans) by counting the number of interventions that occurred over each two minute interval throughout the mission. Two minute intervals provided 10 data points per mission and were used to balance the need to show fine-grain changes in operator behavior with the need to allow a long enough aggregation period to allow the operator to create search tasks and replan. The CHAS model measures the Search Task Rate in tasks created per two minute interval, not tasks/second. Thus, a Sampling Interval parameter is used in the workload module to enable an apples-to-apples comparison between the simulation and experimental data. The Sampling Interval parameter, set

to 120 seconds, is used to convert the search task rate and replan rate back to tasks/second in the calculation of Human Workload (as measured via utilization), shown in Equation 7.

$$Human\ Workload\ (Utilization)$$
$$= \frac{Search\ Task\ Rate * Search\ Task\ Creation\ Time\ Length + Replan\ Rate * Length\ of\ Time\ to\ Replan}{Sampling\ Interval}$$
$$+ NST \tag{7}$$

In order to estimate the Effect of Cognitive Overload on Human Value Added, the model draws upon previous literature in both human supervisory control and SD modeling. It has been established in previous literature that a utilization level over 70% can lead to performance decrements (Cummings & Guerlain, 2007; Nehme, 2009; Rouse, 1983; Schmidt, 1978). Rudolph and Repenning (2002) developed an SD model of the Yerkes-Dodson curve to capture the impact of stress on human decision-making. Their aim was to model the impact of interruptions on the poor decision-making that led to events such as the Tenerife airliner collision or U.S.S. Vincennes disaster. They separated the typical inverted-U Yerkes-Dodson curve into its upward and downward-sloping components. The positive effect of increased stress captured the upward sloping component, including low task load levels, as performance improved with increasing stress levels. The negative effect of increased stress only had an impact on their model outputs once the stress level was high enough to enter the downward-sloping part of the Yerkes-Dodson curve. The negative effect was modeled as a flat line equal to 1 up to the point of overload (thus having no impact on performance as the positive and negative effects were multiplied together), followed by a sharp decline to zero beyond the point of overload (Rudolph & Repenning, 2002).

Inspired by this previous literature, the CHAS model captures the Effect of Cognitive Overload on Human Value Added using a table function, shown below in Figure 25. Up to a workload level of 70%, there is little change in the effect of workload. Above a workload level of 70%, however, the effect would cause a steep drop in Human Value Added up to a maximum workload level of 100%.

This module makes a number of assumptions. First, it assumes that utilization is a good proxy measure for cognitive workload. While utilization has been used in many previous models of operator workload (Nehme, 2009; Schmidt, 1978), it makes two simplifying assumptions: a) all time spent performing tasks is equivalent in terms of the mental resource demand and b) all time

90

spent not performing any active tasks, i.e. monitoring the system, is not drawing on mental resources. Both of these assumptions are dubious, as certainly some tasks require more mental effort than other tasks, and the operator is using some cognitive resources while monitoring the system. Given these limitations, however, utilization has been found to be a useful proxy measure of human cognitive workload in previous studies (Cummings & Guerlain, 2007; Donmez, Nehme, & Cummings, 2010; Schmidt, 1978), especially in monitoring changes in workload over time, and will be used in the CHAS model as a proxy for cognitive workload.
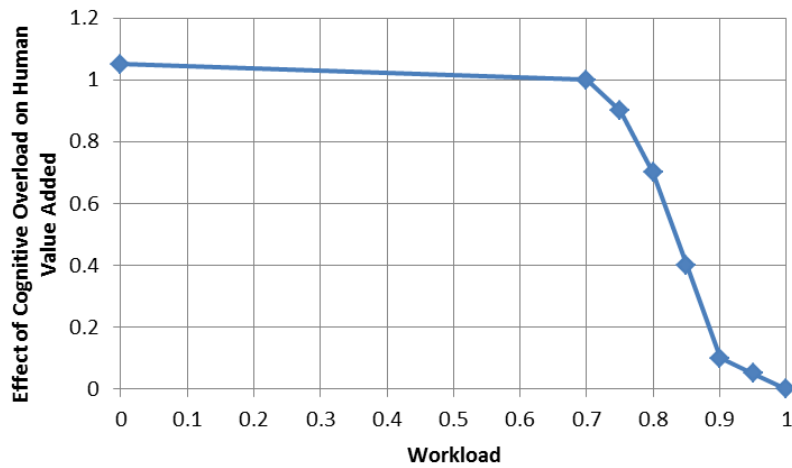
Figure 25. Table function for the Effect of Cognitive Overload on Human Value Added.

Second, the model assumes that the impact of complete cognitive saturation, i.e. 100% utilization, is captured by driving the Human Value Added to the performance of the system to zero. Previous research has shown that under conditions of cognitive overload, perceptual narrowing can occur (Kahneman, 1973). At 100% utilization, the operator will likely begin to miss prompts to conduct important tasks. Other researchers have shown that under conditions of extreme high stress, the typical first response is to freeze (Bracha, 2004). Thus, the model assumes that the Human Value Added to the system will go to zero under complete cognitive saturation. Future research will explore whether in fact the curve should go negative, in that operators begin to make mistakes and detract from system performance under conditions of cognitive overload.

Third, the model assumes that 70% utilization is the point at which performance begins to suffer and that performance drops off according to the curve shown in Figure 25. While the human

supervisory control literature supports the assumption that performance declines beyond 70% utilization, additional empirical evidence supporting the shape of the curve is presented in Chapter 4. Also, sensitivity analysis of how the model outputs change with different workload curves is explored further in Section 6.1.2.

3.5 Feedback Interactions

The CHAS model closes the major feedback loops by relating the rate of interventions and operator workload back to system performance. Human Value Added is calculated by multiplying the Effect of Interventions on Human Value Added by the Effect of Cognitive Overload on Human Value Added. Under normal workload situations (under 70% utilization) the rate of interventions is the key driver of the Human Value Added variable, since the Effect of Cognitive Overload on Human Value Added remains near 1. However, if the operator's cognitive workload reaches too high of a level, there will be a sharp decrease in Human Value Added. This represents the fact that the rate of operator intervention and the effectiveness of these interventions can be in tension, as previous research has shown that high rates of human intervention in a highly automated multi-UV system can lead to worse performance under certain conditions (Clare & Cummings, 2011; Cummings, Clare, et al., 2010).

The interaction of the three main feedback loops shown in Figure 12 is the key driver of performance in a real-time human-automation collaborative scheduling system. First, the Trust in Automation loop is a balancing feedback loop, also known as a negative feedback loop, in that the loop is self-correcting and opposes disturbances. For example, if the Area Coverage Rate were to decrease, the operator's Perceived Present Performance (PPP) would decrease after a time delay, as shown by an arrow connecting the two variables with a plus sign. This would cause an increase in the Perceived Performance Gap (PPG) and a decrease in Perceived Automation Capability. A decrease in Human Trust and an increase in the Search Task Rate would follow. Finally, the Human Value Added would increase and the Area Coverage Rate would return to a higher level. This loop is balancing in that it always seeks to maintain the operator's expected level of performance.

Second, the Expectations Adjustment loop is a reinforcing loop, also known as a positive feedback loop, in that the loop reinforces any disturbance. If the Area Coverage Rate were to be

increase, the operator's PPP would increase, as would the Expected Performance, albeit after a long time delay. This would cause an increase in PPG and a decrease in Perceived Automation Capability. A decrease in Human Trust and an increase in the Search Task Rate would follow. Finally, the Human Value Added would increase and the Area Coverage Rate would continue to increase, reinforcing the original change. This loop tends to reinforce trends towards either increasing or decreasing performance, as the operator's expectations adjust to his or her perception of how well the system is doing.

Third, the Cognitive Overload loop is also a reinforcing loop. If Human Workload were to be increase above 70% utilization, the Effect of Cognitive Overload on Human Value Added would decrease sharply. This would decrease the Human Value Added and therefore decrease the Area Coverage Rate. A decreased PPP, increased PPG, decreased Perceived Automation Capability, decreased Human Trust, and increased Intervention Rate would follow, resulting in an additional increase in Human Workload, reinforcing the trend. This loop works in tension with the Trust in Automation loop, as additional interventions may improve performance, but also cause workload to increase, potentially lowering performance. The loop also captures the fact that there is potentially a tipping point (Rudolph & Repenning, 2002) beyond which the dynamics of the system change dramatically. Once beyond this tipping point, working "harder" to attempt to correct for poor performance can simply lead to further decreases in performance and the need to work "even harder" in a vicious cycle.

3.6 Model Outputs

The CHAS model can aid a designer of future UV systems in predicting the impact of changes in system design and operator training. This can reduce the need for costly and time-consuming human-in-the-loop testing that is typically required to evaluate such changes. It can also allow the designer to explore a wider trade space of system changes than is possible through prototyping or experimentation.

The CHAS model provides a number of output variables that a designer might be interested in capturing. A designer could investigate system performance by analyzing the rate of area coverage or the total area coverage by the end of the mission. Changes in human trust in the AS can be captured, which can be beneficial for the designer to understand, as both undertrust

(Clare, Macbeth, et al., 2012) and overtrust (Parasuraman & Riley, 1997) in automation have been shown to hurt the performance of a system. The rate at which a human operator decides to intervene can be analyzed. This can be important for a system designer to know, for example, to analyze the impact of communications delays between the operator and the vehicles in a decentralized network of UVs (Southern, 2010). The effect of alternative system designs on the workload of the operator can also be captured. In addition, since operator workload is a key driver of human performance, the ability to design a future UV system with an understanding of the impact on operator workload is crucial.

3.7 Model Benefits

Chapter 2 identified a set of research gaps that the CHAS model was designed to address. The following sections describe the benefits of the model in relation to these research gaps. The limitations of the model will be explored in Chapter 6.

3.7.1 Feedback Interaction Among Important Aspects

As was explained in Chapter 2, it is crucial for an effective model of real-time human-automation collaborative scheduling of multiple UVs to capture the feedback relationships among perception, workload, trust, decision-making, and performance. The CHAS model accomplishes this through the three feedback loops that have been implemented in the simulation model: the Trust in Automation loop, the Expectations Adjustment loop, and the Cognitive Overload loop. Rather than treating the aforementioned variables as separate factors, the CHAS model captures the interaction among these components. For example, the tension between the rate of operator interventions and the effectiveness of these interventions due to cognitive overload is explicitly modeled. Components are not static in the CHAS model, but can change over time throughout the simulation of a mission due to feedback. For example, operator expectations of performance, trust, and workload all start at an initial level, but vary over time.

3.7.2 Capturing Impact of Automation Characteristics

The CHAS model captures three crucial details of the automation used in a real-time human-automation scheduling system and the impact of these characteristics on the contributions of the operator and automation to system performance. First, the model explicitly represents the

contribution of the automation to the performance of the system. This provides the system designer with the ability to quantitatively evaluate the impact of improved automation, such as a better search algorithm, on the performance of the system and potentially on the operator's trust level and workload. Second, the model captures the average length of time that an intervention takes. Interventions by the human operator can include replanning to request a new schedule from the AS. If the designer reduces the time that the AS takes to generate a new schedule, the impact on operator workload and performance can be evaluated. Third, the model captures the effect of operator interventions on the human-automation collaboration and thus on system performance. This effect is implemented in the model using an empirically derived, non-linear relationship between the search task rate and human value added that is specific to the automation being used. While it would be difficult to tune this model to a revolutionary system that does not already exist, the CHAS model could potentially allow a system designer to investigate the impact of evolutionary changes to a currently existing system, such as using a different AS, on human and system performance.

3.7.3 Integration of Qualitative Variables

As was defined in Chapter 2, it is critical for an effective model to capture the influence of qualitative variables such as trust or alignment of human and AS goals on system performance. The CHAS model explicitly models human trust and its impact on the rate at which humans intervene into the operations of the team of UVs. The dynamics of trust are captured by enabling trust to adjust over time throughout the mission with some inertia. The alignment of human and AS goals influences both their expectations of how well the system should perform and their perception of the capability of the AS, both of which are captured in the model.

3.8 Chapter Summary

In summary, a System Dynamics model of real-time human-automation collaborative scheduling of multiple UVs has been developed. The System Dynamics modeling process was described and applied to the problem of collaborating with an automated scheduler in a dynamic, uncertain environment. A previous experimental data set was analyzed and led to the creation of a dynamic hypothesis that attempts to explain the differences between high and low performing operators. Using this dynamic hypothesis, the data analysis, and prior literature in human supervisory

control, three major feedback loops were developed. These feedback loops were implemented into a System Dynamics simulation model. This model was specific to the OPS-USERS testbed, and the generalizability of the model will be explored in Chapter 6.

As with any model, a number of assumptions were made in the development of the model. In Chapter 4, these assumptions will be tested as the model is used to simulate the behavior of operators in prior human-automation collaboration experiments. In Chapter 5, the usefulness of the model for predicting the impact of system changes on system performance will be evaluated.

4 Model Validation

This chapter describes the validation process that was conducted for the Collaborative Human-Automation Scheduling (CHAS) model introduced in the previous chapter. While no model will ever be truly validated, as it is a limited, simplified representation of the real world (Sterman, 2000), model testing is essential for building confidence in the soundness and usefulness of the model (Forrester & Senge, 1980). The testing process also attempts to demonstrate whether the model is consistent with the real world that it attempts to capture (Richardson & Pugh, 1981). While a number of methods to validate simulation models have been proposed (Barlas, 1994; Forrester & Senge, 1980; Homer, 1983; Richardson & Pugh, 1981; Sargent, 2005; Sterman, 1987b; Sterman, 2000), the field of System Dynamics (SD) has established a common set of tests for the validation of a simulation model (Forrester & Senge, 1980; Sterman, 2000). This chapter details how each of these confidence-building tests was applied to the CHAS model.

There are three categories of tests that were conducted: model structure tests, model behavior tests, and policy implications tests (Forrester & Senge, 1980). While these tests were used iteratively to refine the CHAS model, the tests presented in this chapter apply to the latest version of the CHAS model. Model structure tests that were conducted and will be discussed in this chapter include: a) boundary adequacy testing, b) dimensional consistency testing, c) extreme conditions testing, d) integration error testing, and e) structure and parameter verification. Model behavior tests include behavior reproduction and sensitivity analysis to parameter changes.

Behavior reproduction testing will be described for three data sets. First, the model was tested on the experimental data which informed construction of the model (Section 3.2). Second, in a "family member" test, data from an experiment that uses the same testbed but under different experimental conditions was used. Third, to test the external validity of the CHAS model, the model was exercised on data from an experiment that uses a different testbed, a multi-robot Urban Search and Rescue simulation (USARSim). Sensitivity analysis included an investigation of the impact of parameter estimate errors on model outputs and Monte Carlo simulations to capture the impact of human variability on model outputs. The final test category, policy

implications, was conducted through a predictive validation experiment and will be described in Chapter 5.

4.1 Model Structure Tests

A number of model structure tests have been conducted on the CHAS model. These tests strive to assess the structure and parameters of the model directly, without necessarily evaluating the relationships between structure and behavior (Forrester & Senge, 1980). The tests described here are: a) boundary adequacy testing, b) dimensional consistency testing, b) extreme conditions testing, d) integration error testing, and e) structure and parameter verification.

4.1.1 Boundary Adequacy Testing

Boundary adequacy tests ask whether the model is appropriate for the purpose for which it was built and whether the model includes all relevant structure (Forrester & Senge, 1980). The primary method of determining the boundary of the model is through inspection of a model boundary chart, shown in Table 2. While all models choose to omit certain concepts and variables in the pursuit of the most parsimonious model, an evaluation of boundary adequacy considers whether there are potentially important feedbacks omitted from the model. Methods of evaluating boundary adequacy commonly include interviews with subject matter experts, review of relevant literature, and assumptions built from experience with the system being modeled.

First, in order to evaluate whether the model is appropriate for the purpose for which it was built, the purpose must be defined. The purpose of the CHAS model was to enable system designers to address the three major challenges that have been identified when human operators collaborate with AS in real-time operations: inappropriate levels of operator trust, high operator workload, and a lack of goal alignment between the operator and AS (Chapter 1). Currently, designers trying to address these issues test different system components, training methods, and interaction modalities through costly human-in-the-loop testing. Through the use of a computational simulation model such as the CHAS model, a designer of future UV systems can simulate the impact of changes in system design and operator training on human and system performance. This can reduce the need for time-consuming human-in-the-loop testing that is typically required

to evaluate such changes. It can also allow the designer to explore a wider trade space of system changes than is possible through prototyping or experimentation.

Next, to evaluate whether the model includes all relevant structure, a model boundary chart was constructed to define which characteristics were captured endogenously, exogenously, or were excluded from the CHAS model boundary (Table 2). The CHAS model was then compared to the six attributes described in Chapter 2 that were important to consider when modeling real-time human-automation collaborative scheduling:

- *Attention Allocation and Situation Awareness*: The operator's attention allocation efficiency was modeled through the Time to Perceived Present Performance (TPPP) time constant, an exogenous parameter. As described in Chapter 3, an operator with higher attention allocation efficiency who was better able to handle the rapid task switching required would have a lower TPPP and detect changes in system performance faster than an operator who struggles with multitasking. The three levels of Situation Awareness were modeled endogenously, by capturing perception of performance (Level I), the perceived performance gap (Level II), and expected performance (Level III).

- *Cognitive workload*: The operator's workload was modeled endogenously, based upon the rate of interventions, the time lengths for these interventions, and the nonscheduling activities that the operator also had to complete. While it is likely that cognitive workload is correlated with attention allocation and situation awareness, it is important to model both attributes in order to capture the feedback interactions among them and the resulting impact on human and system performance.

- *Trust in automation*: The operator's level of trust in the AS was modeled as starting at an initial level defined exogenously, then adjusting endogenously over time to the perceived automation capability with an inertia defined by an exogenous trust change time constant.

- *Human Learning*: The model's calculation of time-delayed adjustments in perceived performance, expected performance, and trust level were all endogenous representations of long-term human learning. Previous versions of the CHAS model also captured how operators learned to use the interface more quickly and efficiently to collaboratively replan with the AS, but this short-term learning component was removed during the model

reduction process (Appendix A). The final model presented in Chapter 3 represents the length of time to replan as a static exogenous parameter.

- *Automation characteristics*: The characteristics of the automation were static during the mission and were thus represented through exogenous parameters and relationships, including automation generated search speed and the relationship between search task interventions and system performance.

- *Human value-added through interventions*: Human value added to system performance was calculated endogenously based on the rate of interventions to coach the sub-optimal automation, while also taking into account the fact that high operator cognitive workload could cause a decrease in the effectiveness of operator interventions, as has been shown in previous research (Clare & Cummings, 2011; Cummings, Clare, et al., 2010).

Table 2. Model boundary chart for the CHAS model.

Endogenous	Exogenous	Excluded
- System performance - Unsearched cells - Searched cells - Probability of a cell being unsearched - Perception of performance (Level I SA) - Perceived performance gap (Level II SA) - Expected performance (Level III SA) - Perceived automation capability - Human trust - Human workload - Search task rate - Replan rate - Human value added	- System characteristics: - Total number of cells - Initial Nonscheduling Task Load (NST) - NST rate of change - Automation characteristics: - Automation generated search speed - Effect of search task rate on human value added to system performance - Human-automation interaction time lengths: - Length of time to replan - Search task creation time length - Initial human conditions: - Expectations of performance - Trust level - Human time constants/delays: - Perceiving performance - Adjusting expectations - Adjusting trust - Non-linear human relationships: - Cognitive overload - Effect of perceived performance gap on perceived automation capability - Effect of human trust on search task rate	- Details of UVs - System or component failures - Vehicle losses - Sensor ranges or types - Communications delays or bandwidth limitations - Allocation of tasks to individual UVs - Safety policies (ex: separation distances) - Human vigilance issues under low workload - Multiple human operators (teamwork & coordination) - Details of the operator control interface - Environmental effects (weather, wind, etc.) - Details of targets - Locations - Movement or evasive maneuvers - Destruction of hostile targets

The majority of these important characteristics were captured endogenously, as they could change throughout a mission. The characteristics of the automation and certain system

characteristics were static during the mission and were thus represented as exogenous parameters. While by definition, initial human conditions (i.e. initial trust level) should be modeled as exogenous parameters, the CHAS model assumed that a number of human characteristics and time constants (i.e. the time constant for adjusting trust) were static and could be modeled as exogenous parameters. Future work should analyze whether these assumptions were valid or whether these characteristics should be modeled endogenously.

The "Excluded" column in Table 2 shows a number of concepts that were excluded to keep the model small enough to capture the major aspects of the system. The CHAS model could potentially be expanded to include many of these other features, such as the impact of vehicle failures, communications delays, safety policies, coordination between multiple operators, or human vigilance issues. These are addressed in the future work section in Chapter 7.

The CHAS model was designed to represent a single operator working on a moderate to high workload mission. The focus of this modeling effort was to capture the perceptions, decisions, and actions of the human operator when working in collaboration with an AS. Thus, the model boundary appears to be adequate to capture the important characteristics of real-time human-automation collaborative scheduling of multiple UVs and to address the challenges that were identified in previous studies.

4.1.2 Dimensional Consistency Testing

Dimensional consistency testing evaluates the units used for each variable and ensures that the units match on each side of every equation. An additional component of dimensional consistency testing is evaluating whether the model includes arbitrary scaling factors that have no real world meaning (Sterman, 2000). All of the equations in the CHAS model passed the dimensional consistency test, both through inspection of all model equations (Appendix C) and through the Vensim$^{®}$ units check function.

4.1.3 Extreme Conditions Testing

Extreme conditions tests are important to evaluate the robustness of a model under extreme inputs. Model outputs should be valid for the entire range of all input variables, not simply at the median input values. If the model still provides valid outputs under extreme conditions, it builds

confidence that the model can be used to extrapolate beyond the data which was used for behavior testing. Extreme conditions tests were carried out in two ways. First, by examining each model equation (Appendix C), one can ask "whether the output of the equation is feasible and reasonable even when each input to the equation takes on its maximum and minimum values" (Sterman, 2000, p. 869).

Second, extreme conditions were imposed on the model through simulations. For example, the Automation Generated Search Speed of the team of UVs was set to an extremely high value (1200% of the normal value) and the model behaves as it should, quickly reaching but not exceeding 100% area coverage (Figure 26). Also, the Automation Generated Search Speed parameter was set to 0, which is representative of the situation where the UVs only tracked previously found targets on their own instead of covering new area. The operator could create more tasks to encourage the UVs to search new area, although system performance would not be as high as if the automation was assisting the operator in the search process (as shown via model simulation in Figure 26). Additional extreme conditions tests are presented in Appendix E, where it was shown that the model behaved appropriately under a variety of extreme conditions.
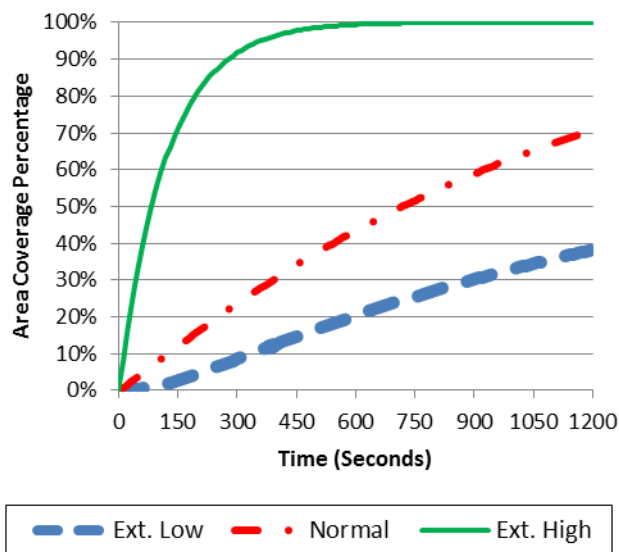
Figure 26. Extreme conditions testing by varying Automation Generated Search Speed.

The CHAS model employs one "artificial" mathematical cap in the calculation of Human Workload to ensure that the model outputs are valid for extreme conditions. As explained in Section 3.4.5, inputs to the workload calculation include the Nonscheduling Task Load (NST) of the operator, such as visual identification of targets and chatting with a command center. Other

inputs to the workload calculation include the rate of search task creation, the replan rate, the average search task creation time, and the average length of time to replan. Without a mathematical cap, it is possible that this calculation could be driven above the maximum workload level of 100% utilization (percent busy time). In the CHAS model, the workload variable is prevented from exceeding 100% through the use of a MIN function. By definition, a quantity such as utilization cannot exceed 100%, thus this artificial means of limiting the workload parameter is necessary for maintaining the validity of model outputs.

4.1.4 Integration Error Testing

As SD models are formulated in continuous time and solved by numerical integration, the selection of integration method and time step are important choices for a SD model. The wrong time step or integration method can introduce unusual dynamics into the results of the model (Sterman, 2000). The time step chosen for the CHAS model was 0.125 seconds, to simulate missions which are typically between 10 to 30 minutes in duration. Humans typically cannot respond to stimuli in less than 0.2 seconds (Wickens & Hollands, 2000), thus the 0.125 second time step should be sufficient for modeling human perception and decision-making. The integration method chosen was Euler. To evaluate whether the model results were sensitive to changes in the time step, the model was run with a time step of 0.0625 seconds, then with a time step of 0.25 seconds. In both cases the results of the model did not change (Appendix E). Also, the model was run with a different integration method and there were no changes in the results.

4.1.5 Structure and Parameter Verification

Structure verification tests compare the structure of the model directly with the structure of the real world that the model attempts to represent (Forrester & Senge, 1980). The main question posed is whether the model and the assumptions that it makes contradict knowledge about the real system. Similarly, parameter verification tests aim to compare the parameters of the model against observations of real life to determine if parameters correspond both conceptually and numerically (Forrester & Senge, 1980). All parameters should have a clear, real-life meaning that preferably can be estimated from actual data, although it is often impossible to directly estimate all parameters in a model from real-world data (Sterman, 2000).

There are two primary methods of structure and parameter verification. First, direct inspection of the model equations can reveal the assumptions made in all causal relationships. Second, experiments can reveal how human operators behave under various circumstances. Data from previous experiments have been used to build the CHAS model and an additional experiment to test some of the model's assumptions will be described in Chapter 5. Throughout the rest of Chapter 4, the model's structure will be evaluated against a variety of data sets.

In terms of structure verification, it was important to verify the interaction between the Trust in Automation and Cognitive Overload feedback loops. As explained in Chapter 3, the model was designed to represent the tension between the positive impact of operator interventions and the effectiveness of those interventions once operator cognitive workload reaches too high of a level. To verify that the structure of the model captures this tension, two simulation runs of the CHAS model are shown in Figure 27. In the "moderate task load" run, the operator's workload remains at a moderate level near 50%, as shown in Figure 27a. In the "high task load" run, the operator's workload is consistently above 75%. The only difference between the two simulations is that Nonscheduling Task load (NST) is set higher for the high task load run. In the real world, this could represent a more frequent need to communicate with a command center or with other operators in the field. It could also represent a system where the operator needs to spend more time analyzing visual images or video to identify or track targets.

The simulations reveal a few interesting results due to this change in task load. First, the area coverage performance of the moderate task load simulation is higher than that of the high task load simulation (Figure 27b). Although performance did not fall dramatically, this result still aligns with previous human supervisory control literature, showing that system performance can suffer when human operators are experiencing cognitive overload (Cummings & Guerlain, 2007; Nehme, 2009; Rouse, 1983; Schmidt, 1978). Second, the model captured the fact that operators will likely detect that the system was performing poorly and attempt to "work harder" to counteract the poor system performance. This is reflected in the increasing search task rate for the high task load simulation (Figure 27c).

104
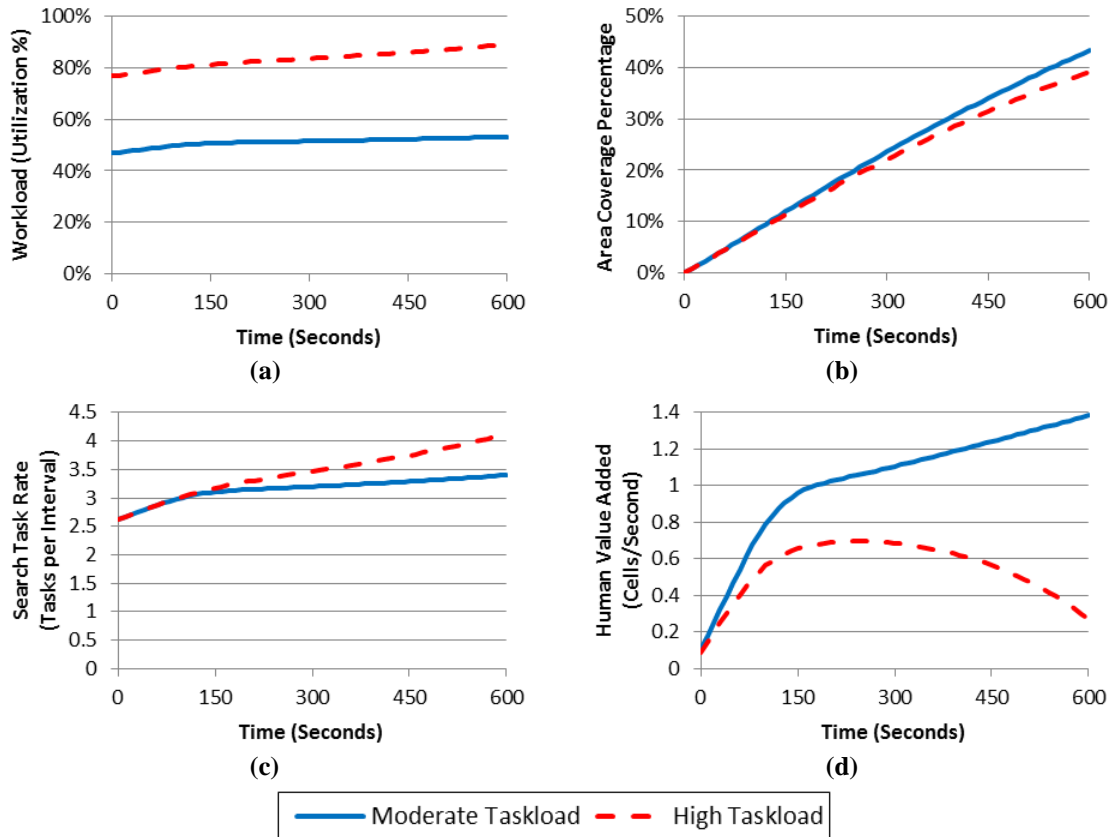
Figure 27. Model simulations of the impact of high task load: a) Workload, b) Area Coverage Performance, c) Search Task Rate, d) Human Value Added.

Third, the model captured the fact that this harder work, creating more search tasks, may be less effective under conditions of cognitive overload. At approximately 150 seconds into the simulation, the negative effects of the Cognitive Overload loop began to override the positive effects of the Trust in Automation loop on Human Value Added to system performance for the high task load simulation (Figure 27d). Normally, a higher search task rate would produce increased Human Value Added, as described in Section 3.4.5. However, at workload levels above 70% utilization, the Cognitive Overload loop becomes active, beginning to reduce the improvement in Human Value Added due to an increasing search task rate. The operator enters a "vicious cycle" where he or she attempts to work harder to fix poor performance, but performance only gets worse because of high workload conditions. Rudolph and Repenning (2002) created a computational model of this tipping point effect with regards to human stress. The tipping point appears to occur at ~80% utilization for the particular simulation shown in Figure 27. Further exploration of this tipping point phenomenon is explored in Section 6.1.2.

Thus, the model structure captures three important aspects of real-time human-automation collaborative scheduling. First, under normal workload conditions, increased operator intervention to guide the sub-optimal automation can benefit system performance. Second, operators are monitoring the system and when they receive feedback that the system is performing poorly they will intervene more frequently in an attempt to improve system performance. Third, under conditions of cognitive overload, system performance will suffer as additional operator interventions will not add value. The impact of high task load on workload and performance will be explored further in Sections 4.2.2 and 4.2.3.

Moving to parameter verification, it was first important to decide how each parameter would be estimated. Table 3 shows a list of all exogenous parameters in the CHAS model. The parameters were divided into four categories based on how they were estimated and whether or not they were allowed to vary between model simulations for a given experiment. First, in the upper left quadrant of Table 3, parameters such as Total Number of Cells were known from the testbed and constant for all operators. The Search Task Creation Time Length was assumed to be constant across operators, as overall variation was small and negligible for the calculation of utilization (Mean: 2.81 s, St. Dev: 1.68 s). Finally, the four non-linear relationships were kept constant between model runs because they were estimated from aggregate data (Section 3.4). Additionally, in the lower left quadrant of Table 3, Automation Generated Search Speed was estimated once via model fitting (Section 3.4.2) and kept constant between model simulations of different operators.

In terms of parameters that were allowed to vary between model runs, the upper right quadrant of Table 3 shows parameters that could be estimated directly from experimental data. The average length of time to replan was estimated directly from experimental data for each operator or group of operators, who were being simulated. Also, the initial NST level and NST rate of change were estimated directly from experimental data via the method shown in Section 3.4.6. Finally, in the lower right quadrant of Table 3, a group of parameters that depend on the human operator that was using the system were estimated via model fitting. For example, each operator potentially had a different initial trust level or a different average ratio of replans to search tasks created. Parameter estimates for model behavior tests are described further in the next section.

Table 3. Exogenous parameters and relationships in the CHAS model.

	Constant between model runs	**Varying between model runs**
Estimated Directly from Experimental Data	- Total Number of Cells - Search Task Creation Time Length - Effect of PPG on Perceived Automation Capability - Effect of Search Tasks on Human Value Added - Effect of Trust on Search Task Rate - Effect of Cognitive Overload on Human Value Added	- Length of Time to Replan - Number of Replans per Search Task - Initial Nonscheduling Task load (NST) - NST Rate of change
Estimated via Model Fitting	Automation Generated Search Speed	- Initial human conditions: - Expectations of Performance - Trust level - Human time constants for: - Perceiving performance - Adjusting expectations - Adjusting trust

4.2 Model Behavior Tests

Model behavior tests include behavior reproduction testing, which will be described for three data sets. First, the model was tested on the initial OPS-USERS data which informed construction of the model. Second, in a "family member" test, data from an experiment that uses the same testbed but under different experimental conditions was used. Third, to test the external validity of the CHAS model, the model was exercised on data from an experiment that uses a different testbed, a multi-robot Urban Search and Rescue simulation (USARSim).

4.2.1 Historical Data Validation

The CHAS model was used to replicate the OPS-USERS experimental data set which was described in Section 3.2. Three sets of data from the experiment were used to evaluate the model's fit to the data: aggregate data for all missions, high performers, and low performers. The high and low performing missions were identified through a cluster analysis described in Section 3.2.

The CHAS model was used to simulate the average behavior of the operators in each of the three groups (low performers, high performers, and aggregate data for all missions). After generally fitting the model to the average behavior of all operators to select an initial starting point for the optimization process, the optimization feature in the Vensim® simulation software was used to fit

the model to the behavior and performance of the three groups. The software used a modified Powell (1964) search to find a local minimum by searching within defined boundaries for each exogenous parameter which was allowed to vary (lower right quadrant in Table 3).

The optimizer evaluated the fit of the model to experimental data for the following variables: area coverage performance, human workload as measured by utilization, search task rate, and replan rate. These four output variables were the only endogenous variables in the CHAS model for which experimental data was available for comparison. Data on the average length of time to replan was used to set the exogenous parameter for this interaction time length. For variables such as trust, expectations of performance, and perceptions of performance, actual data was not available to compare the accuracy of the model simulations. The accuracy of these three variables is evaluated in Chapter 5, where operator ratings of trust, expectations of performance, and perceptions of performance are gathered throughout the mission in a new experiment.

After describing the model fit to the four output variables, a comparison of the model parameter values among the three groups is presented.

4.2.1.1 Model Fit

First, the simulation output for area coverage percentage is compared to average experimental data for each group in Figure 28, with the summary statistics for fit (Sterman, 1984) shown in Table 4. The Theil (1966) inequality statistics provide a decomposition of the error by splitting the Mean Square Error (MSE) into three components: bias (U^M), unequal variation (U^S), and unequal covariation (U^C). The ultimate goal of a model fit is to have small errors between the model and data, with most of the error due to unsystematic, or random, variation. Sterman (2000, pp. 875-877) explains:

> "Bias arises when the model output and data have different means. Unequal variation indicates that the variances of the two series differ. Unequal covariation means the model and data are imperfectly correlated, that is, they differ point to point. Since U^M + U^S + U^C = 1, the inequality statistics provide an easily interpreted breakdown of the sources of error…Ideally, the error (indicated by [Mean Absolute Percent Error] MAPE, [Root Mean Square Error] RMSE, etc.) should be small and unsystematic (concentrated in U^C)."

The simulations had a good fit to the experimental data with coefficient of determination (R^2) values over 0.97 for all three groups, as shown in Figure 28. The model was able to predict the total area coverage performance by the end of the mission within 2.7% for all three groups. The model was most accurate in predicting the low performer group performance curve (Figure 28b), with a percent error at the end of the mission of only 0.8%. In all cases, the largest component of MSE was U^M, as the model underestimated the amount of area coverage for much of the earlier portion of the mission while overestimating the rate of coverage at the end of the mission, for example, overshooting the curve for the all missions group (Figure 28a). This may indicate that that the simplistic system performance module requires refinement for modeling the earlier portions of a mission, although in general the fit is quite good.

Second, the simulation output for human workload is compared to average experimental data for each group in Figure 29 with the summary statistics for fit shown in Table 5. The workload curve for the high performer group (Figure 29c) had the best fit (R^2 of 0.69), replicating some of the non-linear fluctuations in workload that can be seen in the experimental data. The workload curve for the low performer group (Figure 29b) replicated the roughly linear decline in workload seen in the data (R^2 of 0.67). However, for the all missions group, the R^2 value was only 0.50, which is mostly due to the transient workload spike shown in the experimental data at the start of the mission (Figure 29a). This could likely be corrected through the model's representation of NST (Section 3.4.6), which currently under-represents the NST at the start of the mission. Further evaluation of the model's representation of NST is conducted in Section 4.2.2.1 and methods for improving the model of NST are discussed in the future work section of Chapter 7.

In all three groups for the workload fit, the major component of the MSE was unequal covariation, indicating that the error in the model fit to the data was unsystematic. Additional small fluctuations in workload are difficult to capture due to human variability, for example in the amount of time that it takes each operator to create a search task or replan. Capturing this human variability will be discussed further in Section 4.3.2. Additionally, while utilization can serve as a good proxy measure for workload, there are also drawbacks to this measure, as described in Section 3.4.6. While the purpose of this research is not to evaluate utilization as a proxy measure of workload, it should be noted that small fluctuations in utilization may not correspond with actual fluctuations in cognitive workload.

Finally, the two forms of operator interventions, creating new search tasks and replanning, are evaluated together. The simulation output for search task rate is compared to average experimental data for each group in Figure 30, with the summary statistics for fit shown in Table 6. The simulation output for replan rate is compared to average experimental data for each group in Figure 31, with the summary statistics for fit shown in Table 7. First, the model was successfully able to replicate the higher rate of intervention of high performers as compared to low performers. Additionally, the model was able to capture the general trend of increasing intervention as the mission went on. Finally, the model was also able to capture some of the oscillations in intervention frequency that occurred, which were evident in the data for all missions (Figure 30a) as well as the high performer group (Figure 30c).

It should be noted that the fit for search task rate (max R^2 of 0.40) was better than the fit for replan rate (max R^2 of 0.04). This is likely due to the model assumption that the replan rate is directly, linearly related to search task rate. While there is a direct system need, once a search task has been created, to replan in order to assign that new task to the UVs, this is a very simplistic model of operator decisions to replan. The addition of a more sophisticated model of replanning is discussed in the future work section of Chapter 7.

While there is room to improve the fit of the model to the intervention rate data, the most important aspects of the data have been captured, namely the differences in the rates of intervention of the high and low performer groups and the major oscillations in intervention frequency that occurred.

4.2.1.1 Model Parameters

Overall, the model was able to replicate the behavior and performance of the three groups of operators. As described in Section 4.1.5 and Table 3, there were nine possible parameters that were allowed to vary between model runs: Initial Human Trust, Initial Expected Performance (EP), Trust Change Time Constant, Time to Perceive Present Performance (TPPP), Time Horizon for Expected Performance (THEP), Number of Replans per Search Task, Length of Time to Replan, Initial NST, and NST Rate. Six of these variables were modulated by the optimizer, while the other three were estimated directly from the experimental data. The parameter values for each of the three groups are presented in Appendix F.

(a)

(b)

(c)
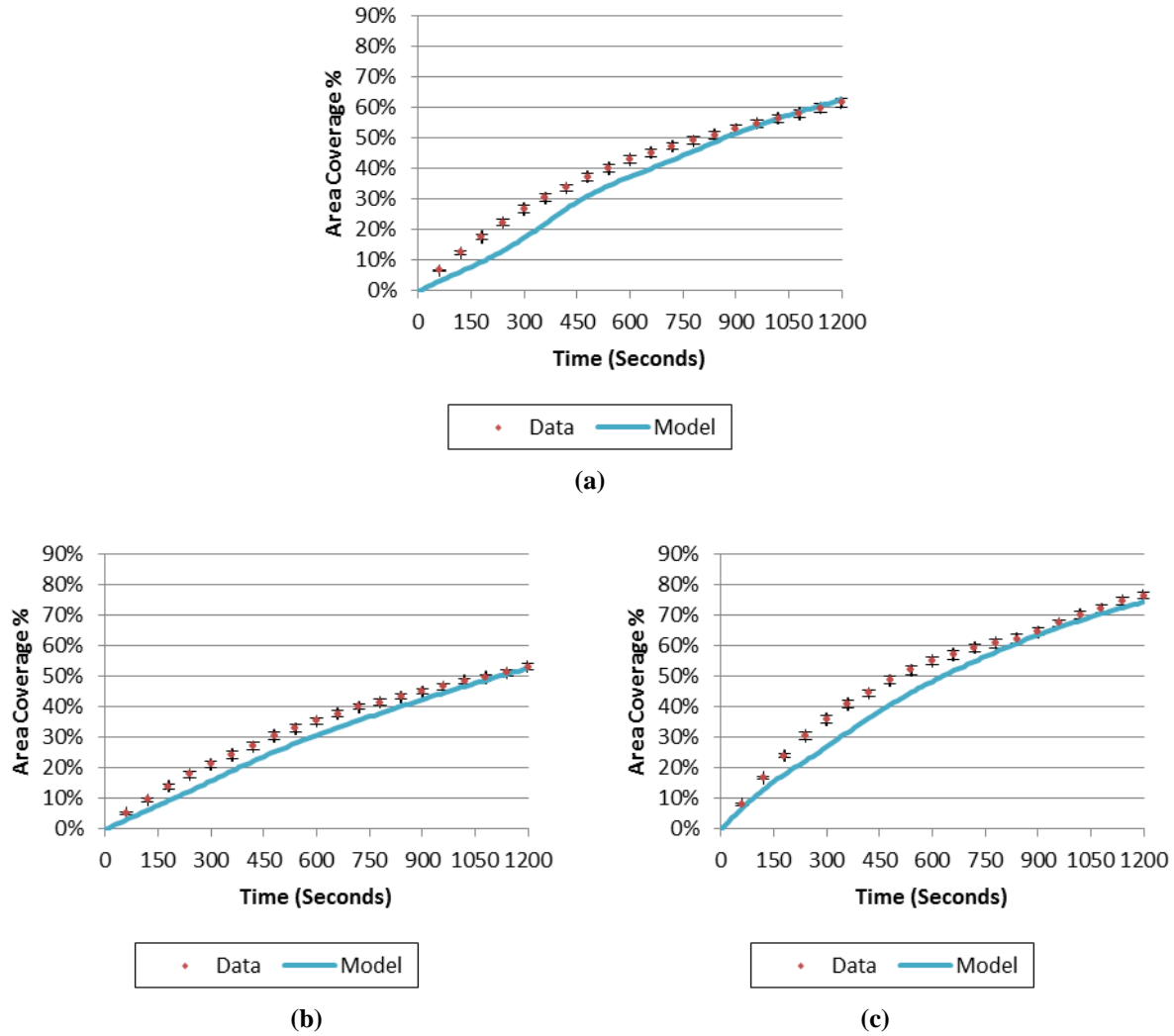
Figure 28. Area Coverage Performance: Simulation vs. Data ±1 SE: a) All Missions, b) Low Performers, c) High Performers.

Table 4. Area Coverage Performance: Simulation to Experimental Data Fit.

Summary Statistics	All Missions	Low Performers	High Performers
Percent Error at End of Mission	1.600%	-0.787%	-2.671%
Coefficient of Determination (R^2)	0.977	0.988	0.981
Root Mean Square Error (RMSE)	0.054	0.039	0.054
Root Mean Square Percent Error (RMSPE)	0.248	0.192	0.148
Mean Absolute Percent Error (MAPE)	0.177	0.152	0.117
Mean Square Error (MSE)	0.003	0.002	0.003
Bias component of MSE (U^M)	0.626	0.802	0.688
Variation component of MSE (U^S)	0.105	0.010	0.013
Covariation component of MSE (U^C)	0.270	0.187	0.299

111

(a)

(b)

(c)
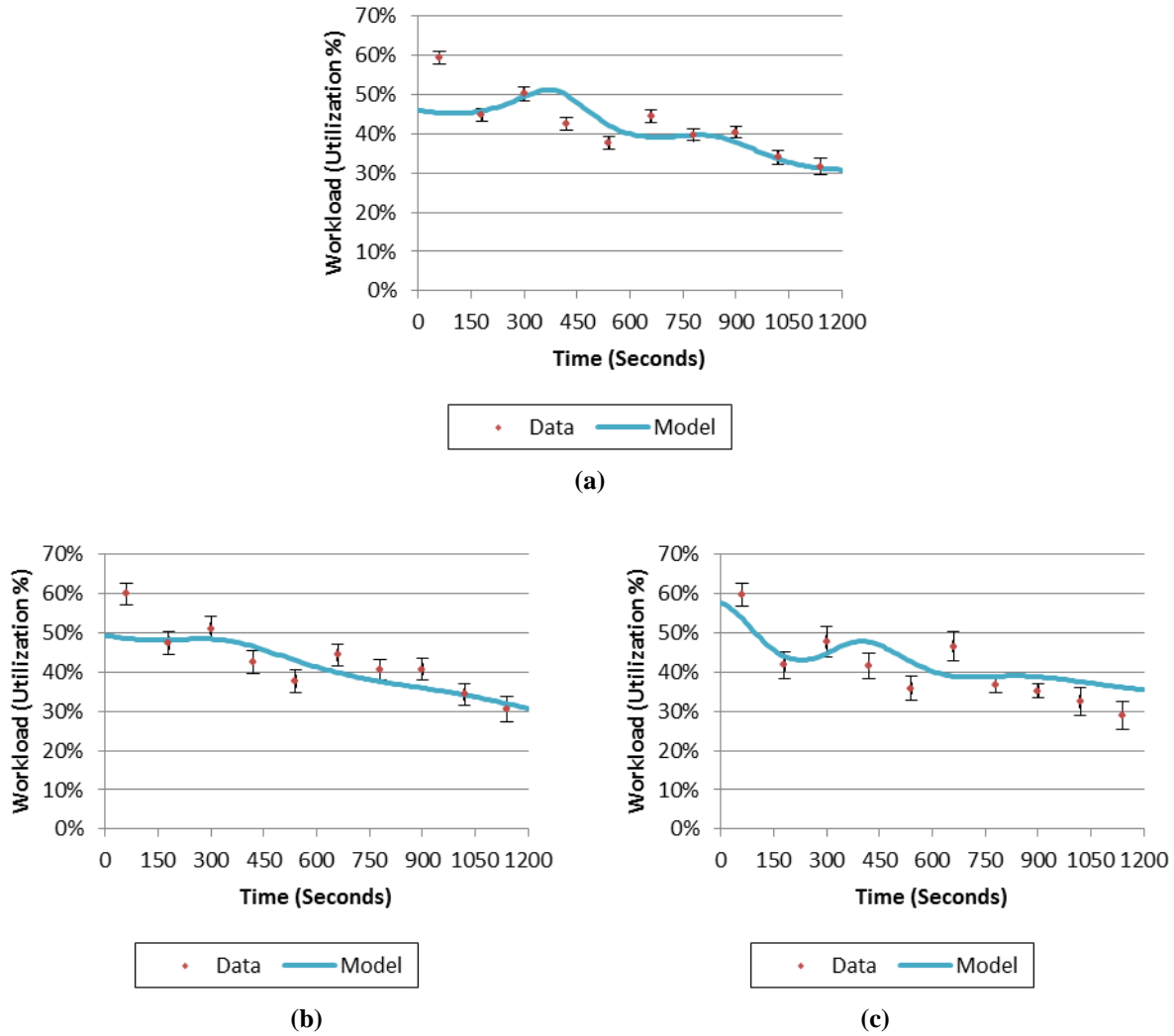
Figure 29. Human Workload: Simulation vs. Data ±1 SE: a) c) All Missions, b) Low Performers, c) High Performers.

Table 5. Human Workload: Simulation to Experimental Data Fit.

Summary Statistics	All Missions	Low Performers	High Performers
Coefficient of Determination (R^2)	0.500	0.676	0.698
Root Mean Square Error (RMSE)	0.055	0.048	0.053
Root Mean Square Percent Error (RMSPE)	0.108	0.099	0.141
Mean Absolute Percent Error (MAPE)	0.077	0.083	0.127
Mean Square Error (MSE)	0.003	0.002	0.003
Bias component of MSE (U^M)	0.036	0.099	0.097
Variation component of MSE (U^S)	0.089	0.168	0.398
Covariation component of MSE (U^C)	0.878	0.734	0.505

112

(a)

(b)

(c)
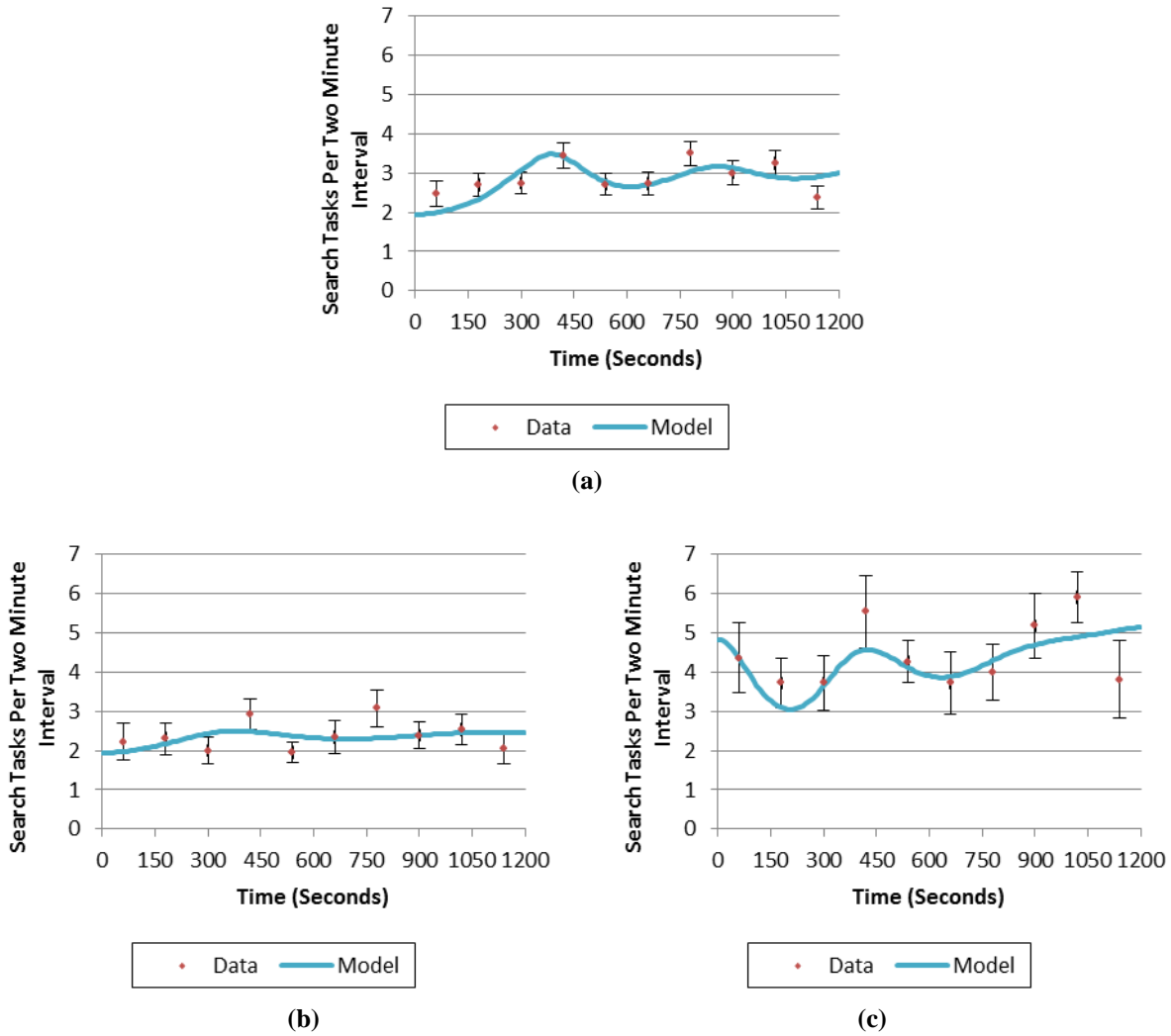
Figure 30. Search Task Rate: Simulation vs. Data ±1 SE: a) All Missions, b) Low Performers, c) High Performers.

Table 6. Search Task Rate: Simulation to Experimental Data Fit.

Summary Statistics	All Missions	Low Performers	High Performers
Coefficient of Determination (R^2)	0.400	0.011	0.351
Root Mean Square Error (RMSE)	0.331	0.375	0.658
Root Mean Square Percent Error (RMSPE)	0.122	0.154	0.147
Mean Absolute Percent Error (MAPE)	0.099	0.126	0.111
Mean Square Error (MSE)	0.109	0.140	0.433
Bias component of MSE (U^M)	0.041	0.013	0.062
Variation component of MSE (U^S)	0.003	0.301	0.101
Covariation component of MSE (U^C)	0.957	0.685	0.837

(a)

(b)

(c)
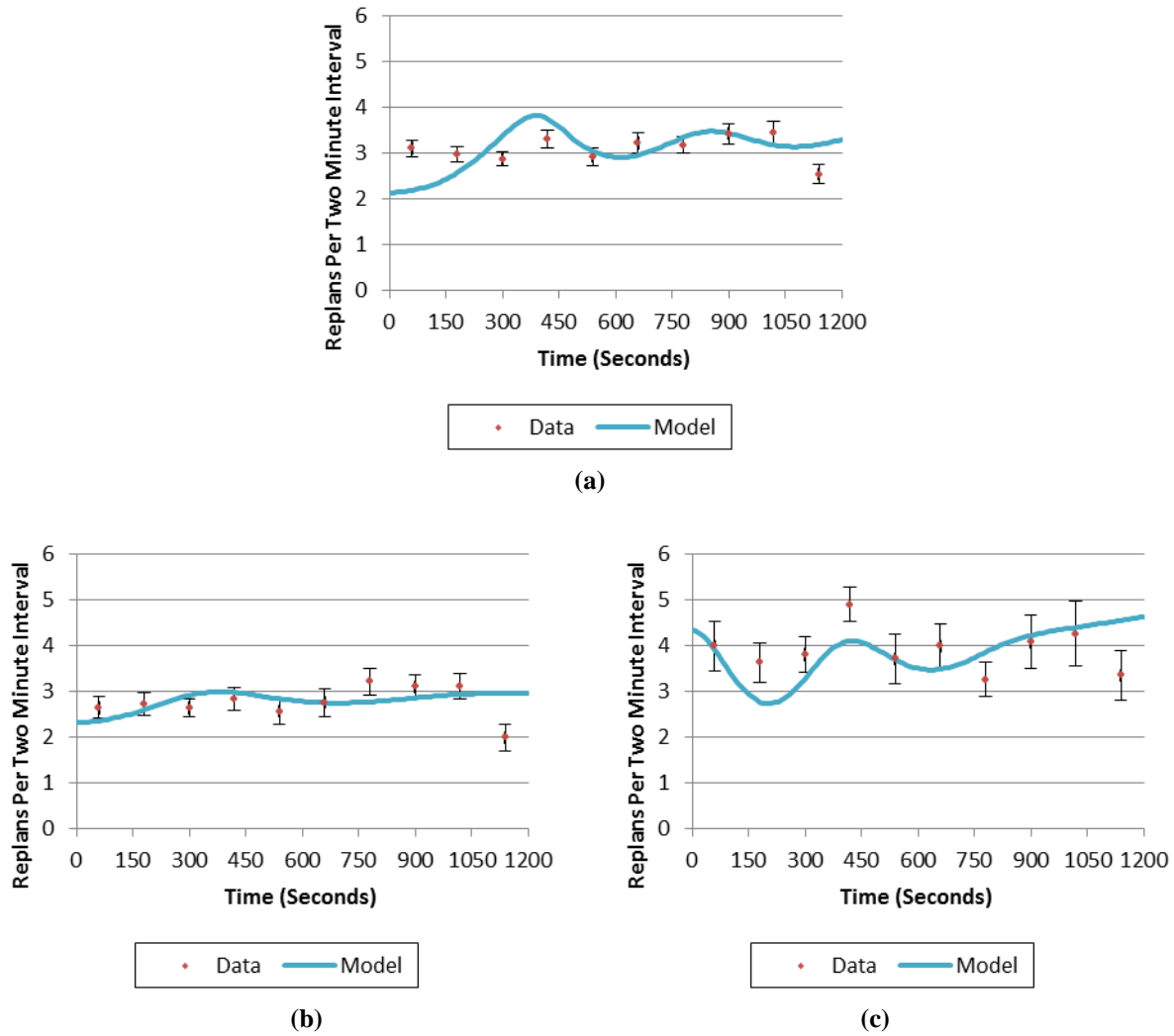
Figure 31. Replan Rate: Simulation vs. Data ±1 SE: a) All Missions, b) Low Performers, c) High Performers.

Table 7. Replan Rate: Simulation to Experimental Data Fit.

Summary Statistics	All Missions	Low Performers	High Performers
Coefficient of Determination (R^2)	0.041	0.001	0.042
Root Mean Square Error (RMSE)	0.452	0.385	0.612
Root Mean Square Percent Error (RMSPE)	0.153	0.173	0.167
Mean Absolute Percent Error (MAPE)	0.125	0.117	0.129
Mean Square Error (MSE)	0.205	0.148	0.374
Bias component of MSE (U^M)	0.000	0.009	0.014
Variation component of MSE (U^S)	0.127	0.158	0.012
Covariation component of MSE (U^C)	0.873	0.834	0.974

114

Once again, for parameters relating to trust, expectations of performance, and perceptions of performance, actual data was not available from the historical data sets to compare the accuracy of the model simulations. The model makes a number of assumptions about how perceptions, expectations, and trust impact more measureable quantities such as intervention rates. The accuracy of these three variables is further evaluated in Chapter 5, where operator ratings of trust, expectations of performance, and perceptions of performance were gathered throughout the mission in a new experiment.

Differences and similarities between the parameter values for each group are compared below:

- *Initial Human Trust*: The model fit process found that operators in the high performer group had the lowest Initial Human Trust, while the low performer group had the highest Initial Human Trust. This corresponds with data analysis from Section 3.2 showing that high performers rated their satisfaction in the AS significantly lower than low performers and had a significantly higher rate of interventions.

- *Initial EP*: The model fit process found that operators in the low performer group had the lowest Initial EP as compared to the high performers and the all missions group. This is reflected in the data analysis from Section 3.2 showing that low performers had significantly higher ratings of satisfaction in the AS and a significantly lower rate of interventions. The model fit assumed that, given the suboptimal automation in the testbed, low performers may have been content with the performance of the system because they had lower expectations of performance to begin with.

- *Time Horizon for Expected Performance (THEP)*: The model fit process found that operators in both the low and high performer groups had a higher THEP compared to the all missions group. This indicates that operators in both groups may have anchored to a certain performance expectation early on and did not adjust that expectation quickly in response to how well the system was actually performing.

- *Time to Perceived Present Performance (TPPP)*: The model fit process found that operators in the low performer group had the highest TPPP. The model assumes that low performers had poorer attention allocation efficiency, had difficulty handling the rapid task switching required, and were slower to detect changes in system performance as compared to high performers.

- *Trust Change Time Constant*: The model fit process also found that operators in the high performer group had the shortest Trust Change Time Constant. This indicates that the model fit assumed that these operators had a smaller amount of trust inertia, enabling them to adjust their level of trust in the AS faster to new information. This is reflected in the sharper oscillations in intervention rate shown in Figure 30c.

- *Number of Replans per Search Task*: High performers had the lowest number of replans per search task, while low performers had the highest ratio of replans to search tasks created. This is reflected in the data analysis from Section 3.2, showing that high performers were intervening more frequently without increasing their workload level, as measured by utilization.

- *Length of Time to Replan*: This parameter was set directly from experimental data, where it was found that high performers had a much lower average length of time to replan.

- *Initial NST and NST Rate*: These parameters were set directly from the experimental data. It was found that operators in all three groups spent the same amount of time attending to nonscheduling tasks.

Overall, these results support the dynamic hypothesis presented in Section 3.3 that high performers were able to anchor to a higher expectation level of performance and adjust to the appropriate level of trust faster. It is likely that they adjusted their level of trust faster through their feedback perceptions of how the system was performing. By adjusting their trust faster, they improved their performance. The oscillation of search task rate and replan rate, seen both in the data and in the model simulations, appears to indicate goal-seeking behavior by the test subjects. The operators learned about the system as they conducted the mission, seeking out the appropriate level of trust and the rate of intervention that produced performance that matched their expectations. The time constants provide an additional level of insight into why high performers did better, as they were able to perceive how the system was performing and learn at a faster rate.

While this section presented the model's ability to capture the average behavior of a group of operators, the model's accuracy at replicating the behavior of all individual missions in this data set was also evaluated (Appendix G). It was found that the mean R^2 value for fitting the model to

all 60 individual missions was 0.968 for the primary performance metric of area coverage and 0.335 for utilization, similar to the fit to the aggregate data presented in this section.

4.2.2 Family Member Validation

The family member validation test evaluates whether a model can generate the behavior observed in other instances of the same system (Sterman, 2000). Thus, the CHAS model was applied to simulate the behavior and performance of operators in a different experiment using the same OPS-USERS testbed discussed in the previous section.

The second data set is from a high task load experiment (Clare & Cummings, 2011), where 31 different operators performed two 10-minute long simulated missions (as opposed to 20-minute long missions in the previous experiment). All operators had a static, *a priori* determined objective function for the AS to use in evaluating schedules and were prompted to view automation-generated schedules at prescribed intervals of either 30 or 45 seconds. Changing the rate of prompts to view new schedules modulates the task load of the operator, such that 30s replan intervals should induce higher workload than the 45s intervals. All operators experienced both replan intervals in a counterbalanced and randomized order. These intervals have been validated in a previous study (Cummings, Clare, et al., 2010). In this experiment (for both replan prompting intervals), there were double the number of targets to find (20 vs. 10) as compared to the previous data set. In addition, the UVs traveled 5 times faster in the high task load experiment and operators received 71% more chat messages that either provided intelligence information updates or Situation Awareness (SA) questions requiring operator response.

A customized version of the CHAS model was developed to model the OPS-USERS testbed with replan prompting at a prescribed interval. The tailored model is shown in Figure 32. There was only one change made to the model as compared to the version of the model presented in Chapter 3. In the Interventions module, as opposed to assuming that the replan rate is directly, linearly related to search task rate, the new model assumes that replan rate is an exogenous parameter. As described in Section 3.2.1, the operator could be prompted to replan when the Replan button turned green and flashed and a "Replan" auditory alert was played. In the experiment described previously and used for model fitting in Section 4.2.1, the operator was prompted to replan when the AS generated a new plan that was better than the current plan.
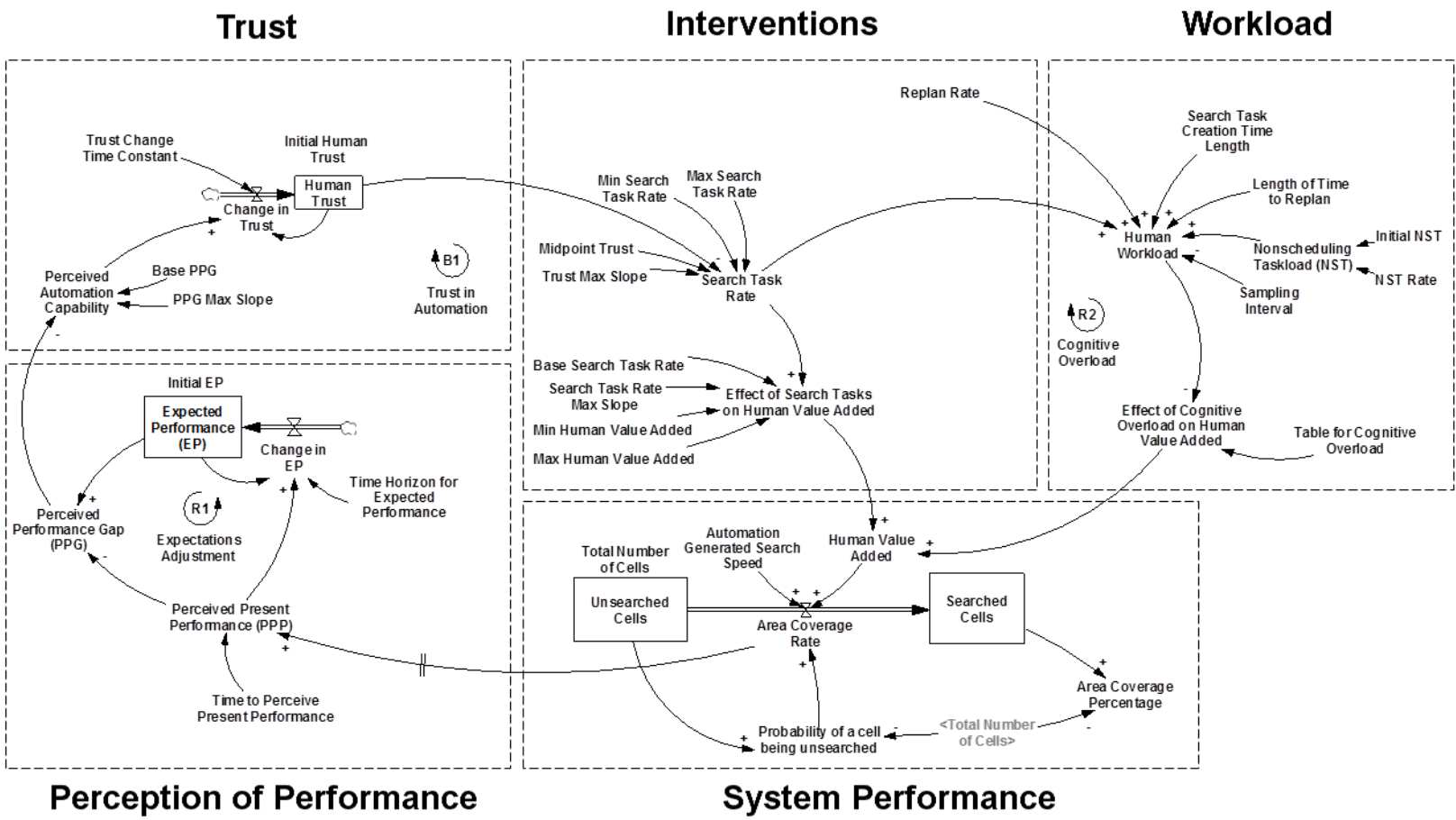
Figure 32. Tailored version of the CHAS model for OPS-USERS with replan prompting at a set interval.

In the high task load experiment presented in this section, operators were prompted to replan at specified time intervals. Thus, modeling the replan rate as an exogenous parameter modulated by the experimental condition is a valid assumption.

Two groups of data from this experiment were used to evaluate the model's fit to the data: the 30 second interval missions and the 45 second interval missions. Both sets of data were used in order to evaluate whether the model could replicate the operator behavior and system performance of each set of missions, given the different replan prompting intervals.

It should be noted that two changes to parameter values (not model structure) were made for this family member validation test. First, the parameters that determine the linear model of Nonscheduling Task Load (NST) were adjusted for this version of the model. The required utilization due to NST was calculated from experimental data from both sets of missions (30 second and 45 second interval), using two-minute intervals, as shown in red in Figure 33. Self-imposed utilization from scheduling activities (creating search tasks and replanning) is shown in blue stripes. The CHAS model representation of NST is shown by the green "model" line. For the 30 second interval missions (Figure 33a), the model represents NST as constant throughout the mission. For the 45 second interval missions (Figure 33b), there is a slight decline in NST over time. The impact of these simplified representations of NST on the model fit to the data is discussed below along with the general evaluation of model fit.
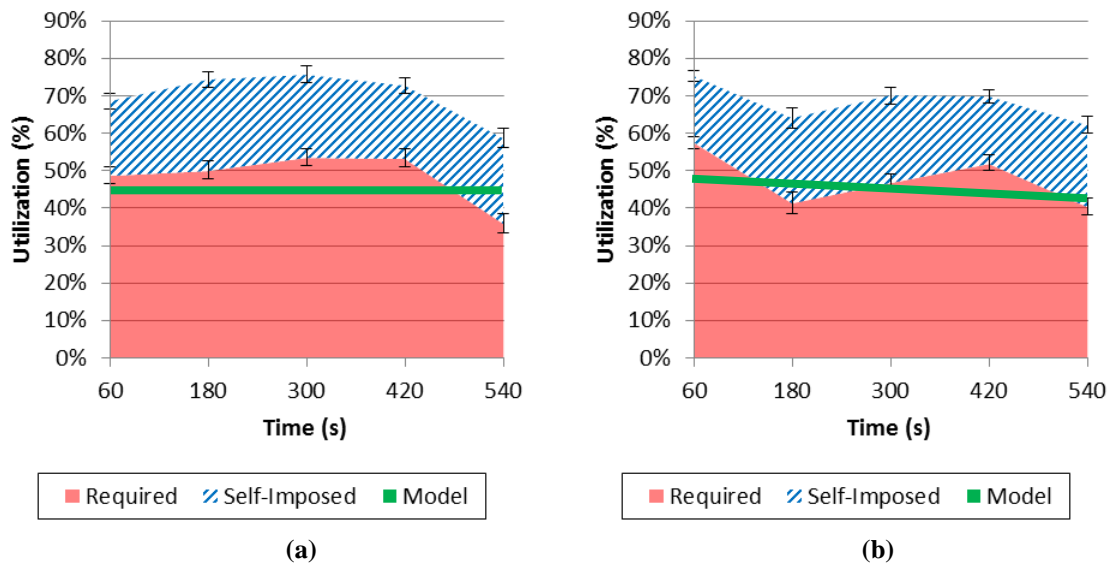
Figure 33. Utilization due to scheduling activities and Nonscheduling Task load (NST). a) 30 second replan prompt interval missions. b) 45 second replan prompt interval missions. Standard error bars are shown.

119

Second, the experiment described in this section had UVs that traveled at increased speeds, thus the Automated Generated Search Speed needed to be estimated. This was done using the same process described in Section 3.4.2 and was found to be 30 cells/second (10 times that of the previous experiment).

The same optimization/fitting process described in Section 4.2.1 was utilized. All estimated parameters for the two sets of missions are presented in Appendix H. After describing the model fit to the four output variables, an evaluation of how well the model replicated the behavior, performance, and workload of these missions is presented.

4.2.2.1 Model Fit

First, the simulation output for area coverage percentage is compared to average experimental data for each set of missions in Figure 34, with the summary statistics for fit (Sterman, 1984) shown in Table 8. The simulations had a good fit to the experimental data with coefficient of determination (R^2) values over 0.93 for both sets of missions. The model was able to predict the total area coverage performance by the end of the mission within 3.3% for both sets of missions. Similar to the model behavior with the previous experimental data set, the model underestimated the amount of area coverage for the earlier portion of the mission while overestimating the rate of coverage at the end of the mission. However, for this data set, the largest component of MSE was U^C for both sets of missions, indicating that the error in the model fit to the data was unsystematic.

Second, the simulation output for human workload is compared to average experimental data for each set of missions in Figure 35, with the summary statistics for fit shown in Table 8. The model was able to capture major differences in workload, such as the higher workload level of the 30 second interval missions as compared to the 45 second interval missions. However, the model fit was not excellent in terms of workload, with R^2 values of 0.37 and 0.36. These R^2 values are comparable to the previous historical data set fit for the all missions group, but worse than the fits for the low and high performer groups (Section 4.2.1.1). The data for the 30 second interval missions (red dots in Figure 35a) shows a slight increase in workload for the first half of the mission, followed by a decline, which is the opposite of the model prediction. The 45 second interval missions data (red dots in Figure 35b) shows an initial decline in workload, followed by

an increase, and then another decline in workload. As will be discussed below, the model's calculation of workload was driven by a simple linear model of NST, while the NST in the actual experiment was more non-linear.

The simulation output for search task rate is compared to average experimental data for each set of missions in Figure 36, with the summary statistics for fit shown in Table 9. The model did an excellent job of replicating the search task rate data, with R^2 values of 0.78 and 0.96. Both the data and the model show a high rate of search task creation initially, declining until approximately halfway through the mission, followed by a sharp increase in the 30 second interval data (Figure 36a) but no increase in the 45 second interval data (Figure 36b).

Finally, the simulation output for replan rate is compared to average experimental data for each set of missions in Figure 37, with the summary statistics for fit shown in Table 9. This version of the CHAS model represents replan rate as an exogenous, static parameter. As expected, replan prompting at an interval of 30 seconds caused a higher replan rate than an interval of 45 seconds (Figure 37). A linear, flat model of replan rate does a decent job of capturing the actual replan rate, with a RMSE of between 0.25 and 0.44 replans per two minute interval, although it does not capture the small oscillations in replan rate shown in the experimental data due to human variability in consenting to replan when prompted (Cummings, Clare, et al., 2010).

4.2.2.1 Evaluation of Behavior, Performance, and Workload Replication

It should be noted that the model replicated the fact that there were no significant differences in performance between the 30 second replan interval and 45 second replan interval missions, while matching the search task rate and replan rate of each set of missions. Also, operators in the 30 second interval missions (Figure 35a) spent almost the entire mission at an average utilization level over 70%. In contrast, operators in the 45 second interval missions (Figure 35b) were only briefly over 70% average utilization at the beginning of the mission, while the rest of the mission was at or below 70% utilization. The fact that the operators in the 30 second interval missions were intervening more frequently, yet did not see any additional benefit to system performance provides additional support for the hypothesis that human interventions can become less effective once the operator's workload reaches too high of a level.
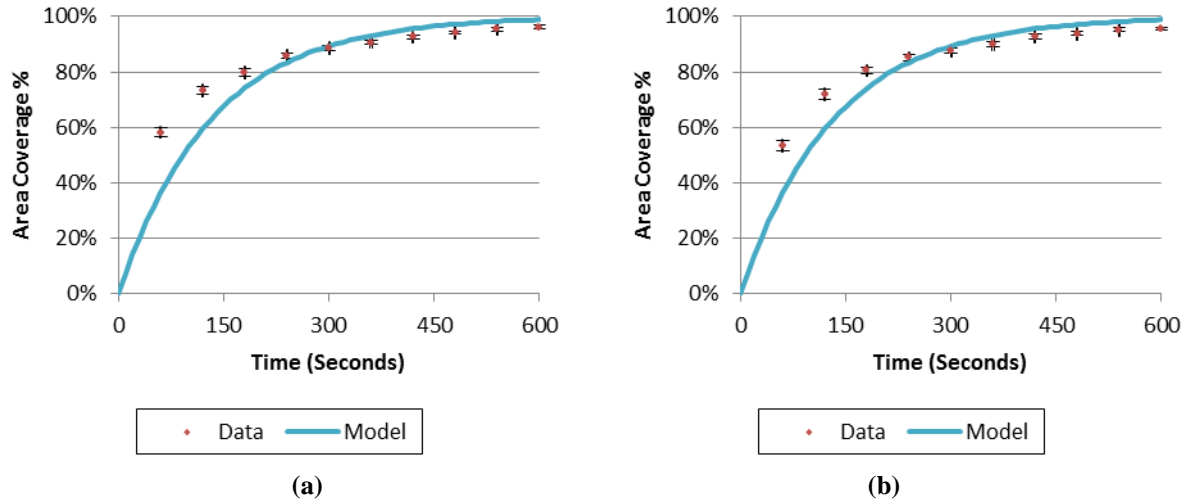
Figure 34. Area Coverage Performance: Simulation vs. Data ±1 SE: a) 30 Second Interval, b) 45 Second Interval.
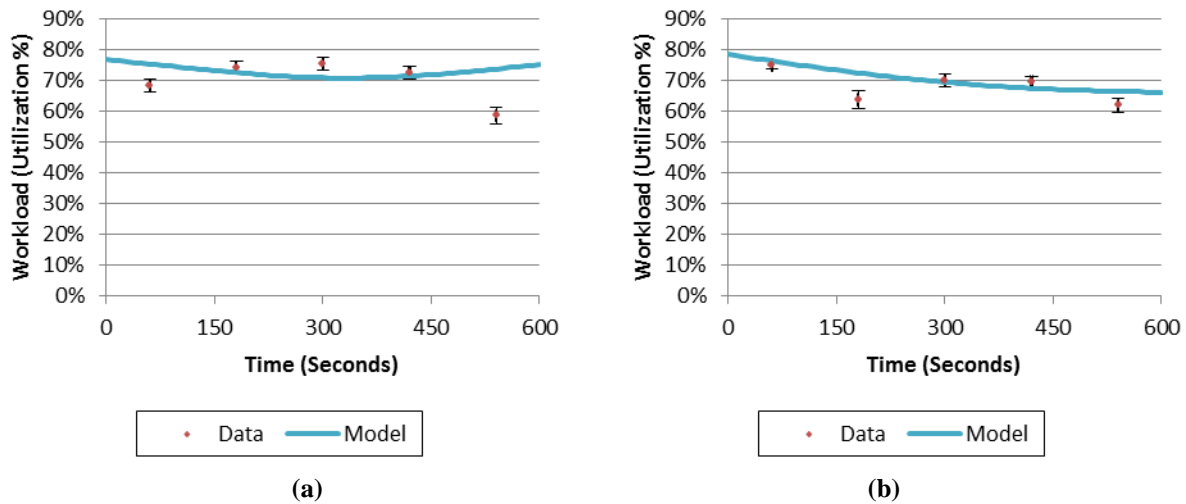
Figure 35. Human Workload: Simulation vs. Data ±1 SE: a) 30 Second Interval, b) 45 Second Interval.

Table 8. High Task load Experiment: Simulation to Experimental Data Fit.

Summary Statistics	30 Second Area Coverage	45 Second Area Coverage	30 Second Workload	45 Second Workload
Percent Error at End of Mission	2.83%	3.28%	N/A	N/A
Coefficient of Determination (R^2)	0.938	0.955	0.372	0.362
Root Mean Square Error (RMSE)	0.085	0.072	0.078	0.045
Root Mean Square Percent Error (RMSPE)	0.135	0.119	0.128	0.070
Mean Absolute Percent Error (MAPE)	0.079	0.074	0.092	0.052
Mean Square Error (MSE)	0.007	0.005	0.006	0.002
Bias component of MSE (U^M)	0.133	0.103	0.143	0.273
Variation component of MSE (U^S)	0.130	0.154	0.332	0.059
Covariation component of MSE (U^C)	0.737	0.743	0.526	0.668
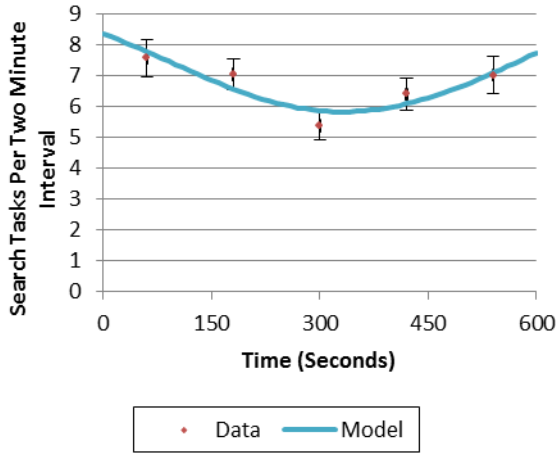
(a)
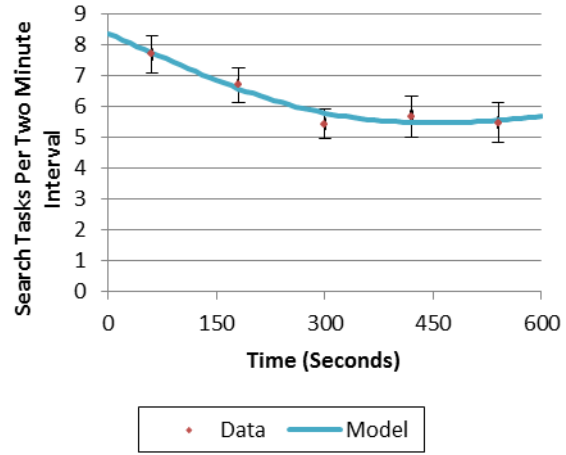
(b)

Figure 36. Search Task Rate: Simulation vs. Data ±1 SE: a) 30 Second Interval, b) 45 Second Interval.
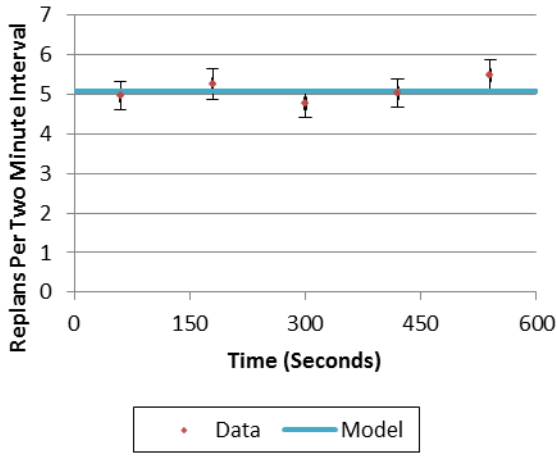
(a)
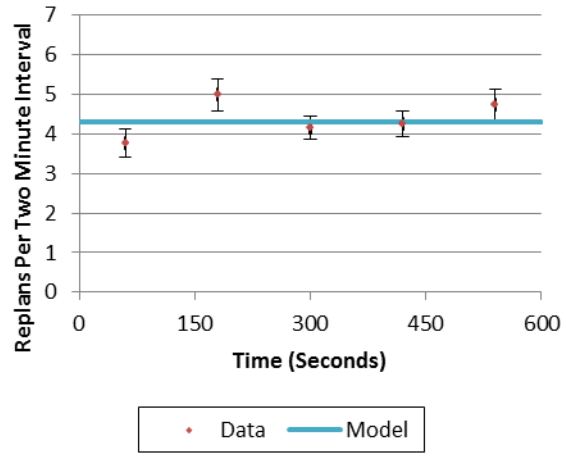
(b)

Figure 37. Replan Rate: Simulation vs. Data ±1 SE: a) 30 Second Interval, b) 45 Second Interval.

Table 9. High Task load Experiment: Simulation to Experimental Data Fit.

Summary Statistics	30 Second Search Task Rate	45 Second Search Task Rate	30 Second Replan Rate	45 Second Replan Rate
Coefficient of Determination (R^2)	0.781	0.956	0.000	0.0000
Root Mean Square Error (RMSE)	0.353	0.191	0.245	0.443
Root Mean Square Percent Error (RMSPE)	0.057	0.034	0.048	0.109
Mean Absolute Percent Error (MAPE)	0.050	0.027	0.042	0.083
Mean Square Error (MSE)	0.125	0.036	0.060	0.197
Bias component of MSE (U^M)	0.000	0.048	0.000	0.039
Variation component of MSE (U^S)	0.024	0.021	1.000	0.961
Covariation component of MSE (U^C)	0.976	0.931	0.000	0.000

The model's predictions of system performance were accurate because of the interaction of two feedback loops. First, the Trust in Automation loop provided a positive effect on Human Value Added for the higher rate of creating search tasks in the 30 second interval missions. However, the model also calculates an "Effect of Cognitive Overload on Human Value Added" based on the workload level of the operator (Section 3.4.6). This effect is a non-dimensional variable which scales the Human Value Added to system performance, thus a lower Effect of Cognitive Overload on Human Value Added will reduce the Human Value Added. In the model simulations of the 30 second interval missions, the Cognitive Overload loop produced a lower Effect of Cognitive Overload on Human Value Added, as shown by the red dashed line which is mostly below the blue line in Figure 38. This occurred because operators in the 30 second interval missions spent almost the entire mission above 70% utilization. Both lines in Figure 38 initially increase because the search task rate for both sets of missions decreased, causing a decrease in workload for the first half of the mission. However, the increasing search task rate of the 30 second interval missions in the second half of the mission caused the simulation to calculate increasing workload, thus the decrease in the red dashed line.
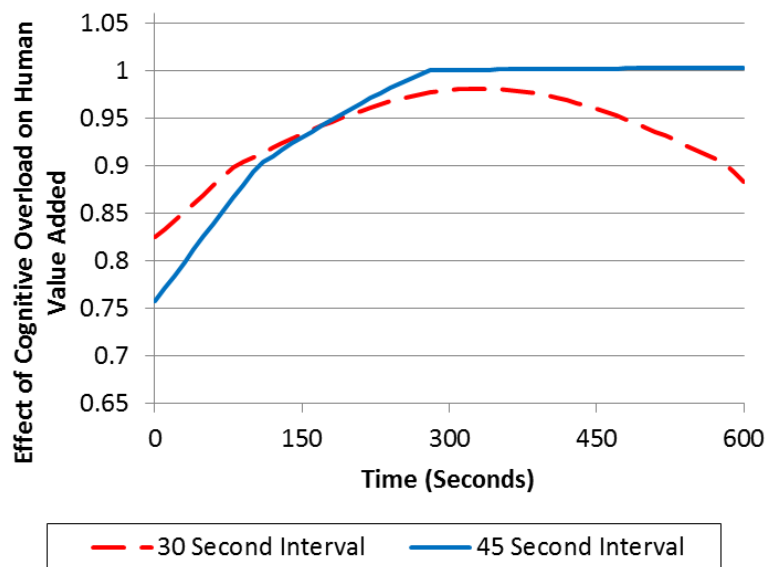
Figure 38. Model simulation of the impact of cognitive overload on performance.

This experimental data set provides supporting evidence for the structure of the CHAS model, demonstrating that high workload can potentially reduce the effectiveness of operator interventions in real-time human-automation collaborative scheduling of multiple UVs. The

124

CHAS model was able to replicate the intervention behavior and system performance of these two sets of missions by effectively modeling the impact of high workload.

While the model was able to capture major differences in workload, such as the higher workload level of the 30 second interval missions as compared to the 45 second interval missions, the model's replication of workload fluctuations throughout the mission was subpar. The model did not capture the non-linear changes in workload shown previously in Figure 35. One major reason for the poor fit to the workload experimental data is that the CHAS model uses simple, linear models of NST, as shown previously in Figure 33. Both of the simplified linear representations of NST did not accurately capture the fluctuations in NST. To more accurately capture the impact of NST on workload, however, would require a non-linear implementation of NST, which could replicate the rise and fall of NST over time. Alternatively, the model could be driven with external data that contains accurate NST information for each mission. Both of these potential concepts are discussed in the Future Work section of Chapter 7.

It should be noted that the NST for this high task load experiment was different from the NST in the previously described OPS-USERS experiment. For the high task load experiment, there were 71% more chat messages and double the number of targets to identify, leading to required utilization due to NST of roughly 45% (Figure 33). In the previously described OPS-USERS experiment, required utilization due to NST was substantially lower, generally ranging from 10-30% (Figure 24).

Comparing the fit of the CHAS model between the original OPS-USERS experiment (Section 4.2.1) and the high task load experiment presented in this section, in both cases the area coverage performance fits were excellent, with R^2 values above 0.93. Generally, the model replicated the intervention differences between the various groups in each experimental data set. The fit was best for the search task rate in the high task load experiment, with R^2 values above 0.78. Finally, in terms of workload, the CHAS model replicated the workload curves of the original OPS-USERS experiment with R^2 values between 0.5 and 0.69. However, for the high task load experiment, the workload goodness of fit values were lower (0.36-0.39). This is due to the model's simplified representation of NST. While the workload calculation could be refined,

overall, it appears that the CHAS model was successfully able to replicate the intervention behavior and performance of two OPS-USERS data sets.

4.2.3 External Validation

While the CHAS model has been tested on the original data set that was used to inform construction of the model and an additional data set from a different experiment that used the same testbed, it was important to evaluate the CHAS model on a data set from a different testbed. This would provide a test of the external validity of the CHAS model, how well it could generalize to other real-time human-automation collaborative scheduling systems. Thus, the CHAS model was applied to simulate the behavior and performance of operators in a multi-robot Urban Search and Rescue experiment.

Researchers have been exploring the use of autonomous robots for Urban Search and Rescue (USAR) for over a decade. Robots can go to places that are impossible or too dangerous for human rescuers. In urban search and rescue, robots usually need to navigate through a complex environment to map the environment and look for victims. Currently in practice, two operators are usually required to manually control a rescue robot. With autonomous path planning and scheduling algorithms, it is possible to reduce the workload of operators and increase the number of robots they can control.

In order to apply the CHAS model to a USAR mission, data was collected from a previous USAR experiment (Gao, Cummings, & Bertuccelli, 2012) conducted using a 3-D simulation testbed based on USARSim (Lewis, Wang, & Hughes, 2007). The task of the operators was to monitor the cameras of robots and mark the positions of victims on the map when they appeared in the cameras (Figure 39). The goal was to mark as many victims as possible correctly. By default, robots searched autonomously for victims based on plans and paths developed by an AS. Operators could choose to take manual control and teleoperate an individual robot during this process when they felt it was necessary. Operators worked in teams of two to monitor a total of 24 robots. Each team went through three trials of 25 minutes. In these three trials, the building maps were the same, but the locations of the 34 victims were different. At the end of each trial, subjective workload ratings were obtained from each operator using the NASA-TLX scale (Hart, 1988), which measures six sub-dimensions of workload.

126

Analysis of data from this experiment is presented next, which informed the customization of the CHAS model to model this testbed. Later, the model fit to the data is described, along with an evaluation of how well the model replicated the behavior and performance of different groups of operators.
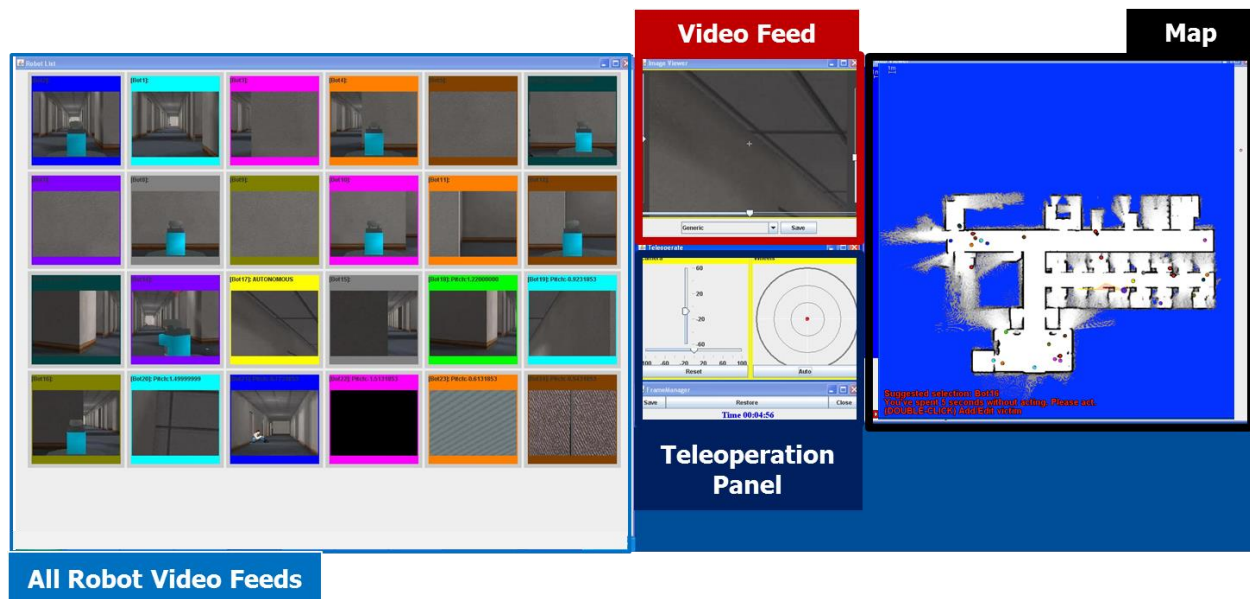
Figure 39. USARSim interface for controlling multiple search and rescue robots.

4.2.3.1 Data Analysis

Analysis of the data from this experiment first showed that operators spent a longer time on teleoperation in later trials than in earlier ones, as shown in Figure 40. ANOVA analysis shows that this impact of trial sequence on the time spent on teleoperation is significant ($F(2,141)=7.37$, $p<0.001$). In interviews after the experiment, many participants said that the AS did not do a very good job and could not be trusted. They stated that robots often went back to places already explored, sometimes multiple times, while leaving some other places unexplored. They complained that the search was not thorough if they relied on only the AS. Even though intervening via teleoperation requires more effort than just relying on the AS, operators chose to teleoperate when the robots were not going where they wanted them to go.

Thus, both quantitative and qualitative data indicate that operators lost trust in the automation throughout the three trials and chose to intervene more frequently. This indicates a link between the operator's perception of the automation's capability, operator trust, and the amount of

teleoperation conducted by the operator. The data also show that trust can change both during and in-between missions. Both of these findings support assumptions in the CHAS model.
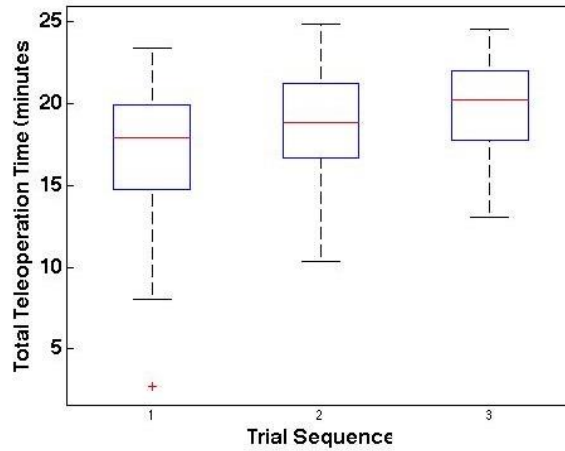
Figure 40: USAR Experiment: Total Teleoperation Time versus Trial Sequence.

In a second analysis, for each 25-minute experiment trial, the number of teleoperation "actions" per minute was counted. To smooth out short-term fluctuations in the time series data, the moving average over each five-minute period was calculated, resulting in 21 data points per trial. Hierarchical clustering was used to classify each of the 144 experiment trials into groups. The goal was to identify groups of operators who had distinct behavior in terms of the frequency of teleoperation.

This analysis identified six distinct clusters of trials. The first three clusters contained only seven trials in total, and were removed from further analysis. The last three clusters had different levels of teleoperation as shown in Figure 41a. The trials in Cluster 4 (named Low TeleOp), shown in red squares in Figure 41a, had the lowest frequency of teleoperation. These trials also had significantly worse performance $((F(2,134)=16.67, p<0.001)$, in terms of total victims found, than the other two clusters, as shown in Figure 41b. Thus there is a positive relationship between decreased teleoperation frequency in this experiment and decreased performance. Combined with the previously discussed link between trust and the frequency of teleoperation, it indicates an indirect, but crucial relationship between operator trust in automation and system performance.

It should be noted, however, that while the trials in Cluster 6 (named High TeleOp) had significantly more teleoperations than those in Cluster 5 (named Med TeleOp) (Figure 41a),

128

there were no statistically significant differences in system performance between these two groups (Figure 41b). There appear to be diminishing returns in terms of performance with more teleoperation.
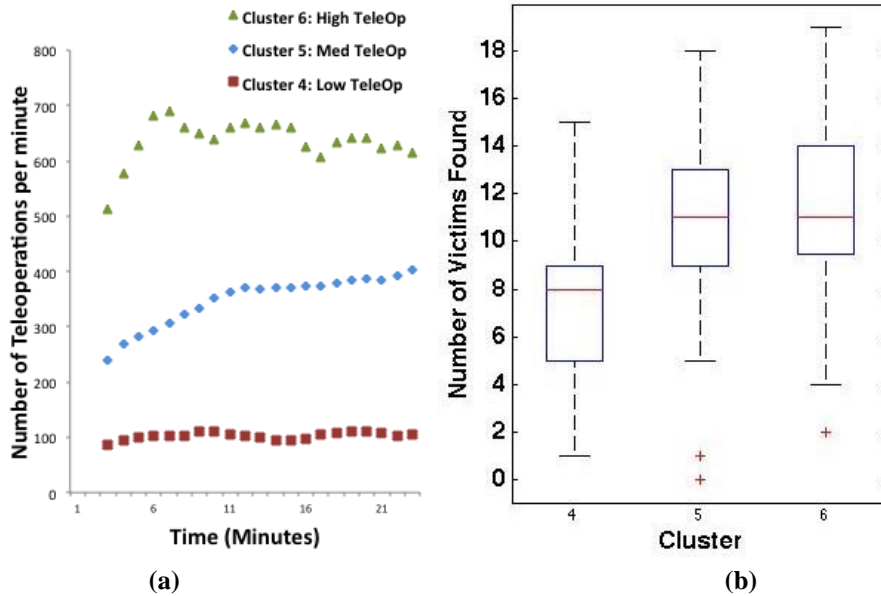
Figure 41: (a) Average teleoperation frequency versus mission time; (b) Number of victims found by teleoperation cluster.

Finally, in order to investigate whether the frequency of teleoperation is related to operator workload, an analysis of the differences in NASA-TLX workload ratings among the three clusters was conducted. ANOVA analysis showed that there were significant differences on the temporal demand dimension of workload between clusters ($F(2,134)=68.37$, $p<0.001$), but not on overall workload ratings or any other dimensions. Operators in the High TeleOp group reported higher temporal demand, which indicates that they felt increased time pressure and that the pace of the simulation was rapid. This indicates that operator workload is related to the number of operator interventions, supporting another key CHAS model assumption.

4.2.3.2   Model Customization

In order to inform the customization of the CHAS model for simulating human control of multiple robots for an USAR mission, the OPS-USERS and USARSim testbeds were compared. The two testbeds are similar in a number of ways. They both enable operators to collaborate with an automated scheduling and path planning algorithm for controlling multiple robotic vehicles under *goal-based* control. As described in Chapter 2, in *goal-based* control, the human operator

guides the high-level goals of the team of UVs (as opposed to guiding each individual vehicle) and the AS assumes the bulk of computation for path planning and task assignment optimization. The USARSim testbed typically simulates homogeneous ground-based UVs, while the OPS-USERS testbed can control heterogeneous UVs (air, land, sea). However, in the OPS-USERS testbed, the human operator does not need to worry about the distinction between different types of UVs, as the automation checks the feasibility of all task assignments based on the capabilities of the UVs. Also, both testbeds utilize a Map Display with a top-down view of the world.

There are four major differences between the two testbeds. First, the goals of the missions are slightly different. In the OPS-USERS testbed, the goal is to find and track moving ground targets over a large, but fairly open area. In the USARSim testbed, the goal is to find and mark stationary human victims inside a building with an unknown layout. Despite this difference, however, both are search missions for a fixed number of victims/targets over a fixed area. Second, in OPS-USERS, the main interventions that the operator can use to coach the automation are the creation of search tasks and replanning. The operator cannot manually control any particular UV in OPS-USERS. In contrast, the USARSim testbed allows operators to manually teleoperate a single robot at a time. This is the only intervention that the operator can use, both to mark victims and to guide the search process.

The third difference between the two testbeds is that the OPS-USERS testbed requires operators to multi-task, by monitoring the UVs, responding to chat messages, classifying visual images, and interacting with the AS to generate new schedules. In contrast, the USARSim testbed only requires operators to focus on the primary objective of monitoring the robots and marking victims. However, the workload level of operators in USARSim is just as high, if not higher than in OPS-USERS because there are 24 robots as compared to UVs in OPS-USERS. The final major difference between OPS-USERS and USARSim is that OPS-USERS is purely a single-operator simulation, while the USARSim experiment described here was a two-player, team activity, where operators collaborated to control the group of robots. Although the USARSim data set comes from a team-based experiment, this modeling effort focused on the concept of a single operator controlling multiple robots, thus team coordination and task allocation among multiple operators was not considered.

130

Given these similarities and differences between OPS-USERS and USARSim and the analysis of data from the USARSim experiment, a customized version of the CHAS model was developed to model human-automation collaborative scheduling for a USAR mission. The tailored model is shown in Figure 42. There were three major changes to the model from the version which has previously been presented. The first change was simply a terminology and units change. The system performance module was customized to the USAR mission with a focus on finding victims. In the USARSim experiment, once a victim is visited, it is marked as "found", which is the primary performance metric. As more and more victims are found, the likelihood that a new victim is found declines, which lowers the Victim Discovery Rate. Although the data set comes from a team-based experiment, this model focuses on the concept of a single operator controlling multiple robots, thus team coordination and task allocation among multiple operators is not considered. Thus, the Total Number of Victims parameter was set to 17, half of the actual total in the experiment, assuming an equal division of labor between the two operators.

The second change to the model was that the main operator intervention was changed to the Number of Teleoperations. This was modeled in the same manner as the Search Task Rate in the previous CHAS model, where the Number of Teleoperations is negatively non-linearly dependent on Human Trust level (higher trust means less likely to intervene) using a logit function. The model assumes, just as before, that a higher number of teleoperations improves the value that the human operator adds to the rate of finding victims. While the change itself was only a terminology change and there was no change to the underlying model structure, the parameters that define the non-linear relationships had to be estimated from experimental data.

The third change to the model did impact the model structure. Replan Rate was removed from the model, as there was no separate replanning process in the USARSim testbed. Additionally, the workload module was simplified by scaling the Number of Teleoperations directly to workload using an Effect of Teleoperations on Workload parameter. Utilization data for each operator was not available in the data set for model fitting. However, using data on the total teleoperation time (such as that shown in Figure 40), the fraction of the mission spent in manual control of the robots could be used as a proxy for utilization to enable estimation of the Effect of Teleoperations on Workload parameter. While this is an oversimplification of the workload calculation, the impact of this assumption on model results will be discussed below.
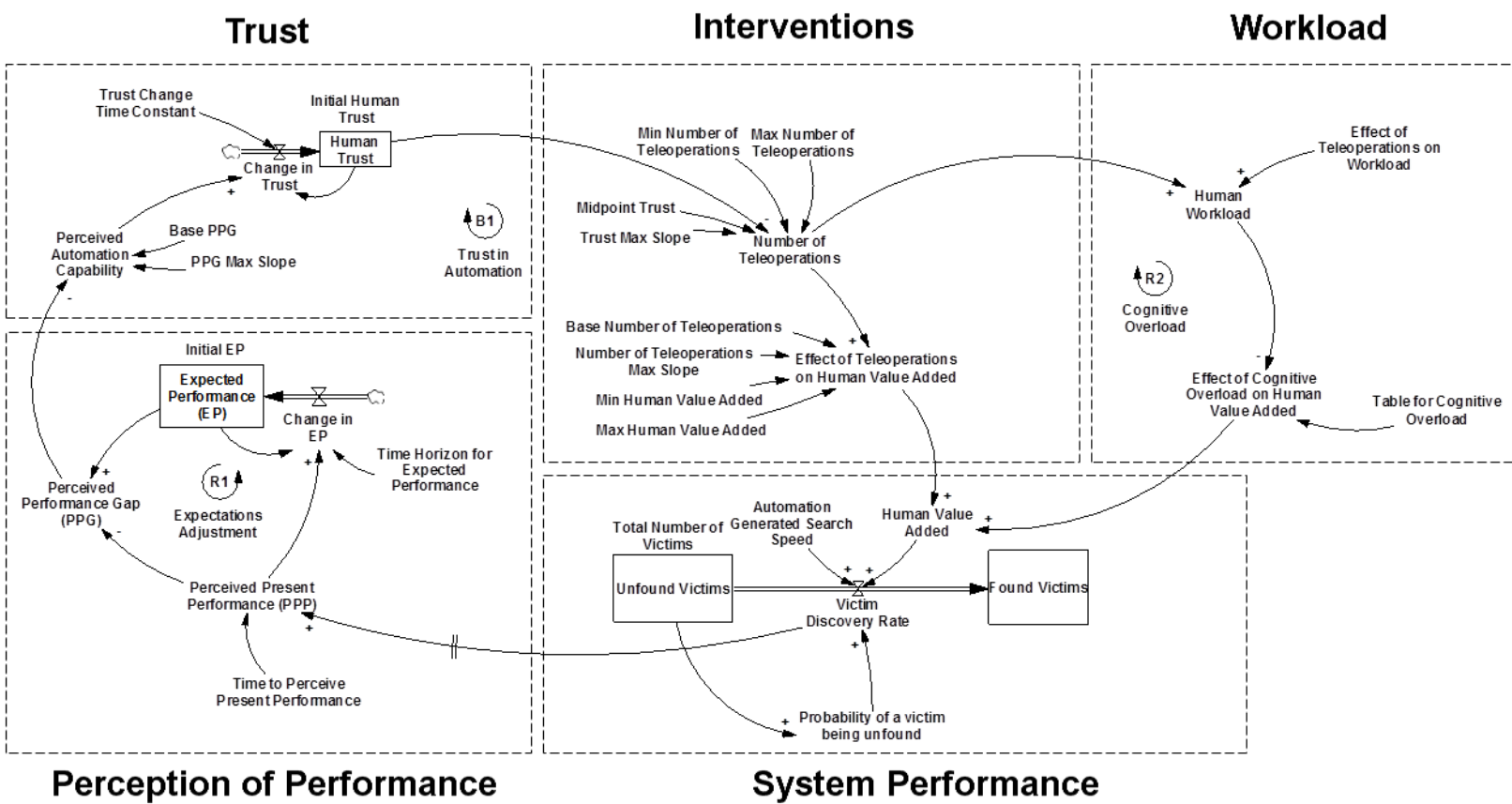
Figure 42. Tailored version of the CHAS model for a multi-robot USAR mission.

The three teleoperation cluster groups described in the previous section were used to evaluate the model's fit to the data. The data available for fitting were the Number of Teleoperations and the number of Found Victims over time. The same optimization/fitting process described in Section 4.2.1 was utilized. All estimated parameters for the three groups are presented in Appendix I. After describing the model fit to the two output variables, an evaluation of how well the model replicated the behavior and performance of these missions is presented.

4.2.3.3 Model Fit

First, the simulation output for Number of Teleoperations is compared to average experimental data for each group in Figure 43, with the summary statistics for fit shown in Table 10. The model had a good fit to the Number of Teleoperations data. For the Medium TeleOp and High TeleOp groups, the R^2 values were 0.96 and 0.80, as the model was able to replicate the initial increase in Number of Teleoperations, followed by a slower adjustment over time. The experimental data for the High TeleOp group (Figure 43) appears to show overshoot of a desired Number of Teleoperations, followed by a slow decline in teleoperations with small oscillations.

The model captured the overshoot and decline to a desired intervention level shown by the High TeleOp group (Figure 43), but did not capture the small oscillations. Overall, this data suggests that operators were likely learning about the system as they conducted the mission, seeking out the appropriate level of trust and the rate of intervention that would produce performance that matched their expectations. High performers appear to have learned quickly and adjusted their trust level more dramatically, as shown by their rapid increase in Number of Teleoperations.

While it may appear that Low TeleOp group had a subpar fit, with a R^2 value of only 0.23, recall that R^2 is a measure of the proportion of the variation in the experimental data explained by the model. As there was little variation over time in the frequency of teleoperation for this group in the experimental data, the model cannot and should not recreate these small variations in order to avoid overfitting.
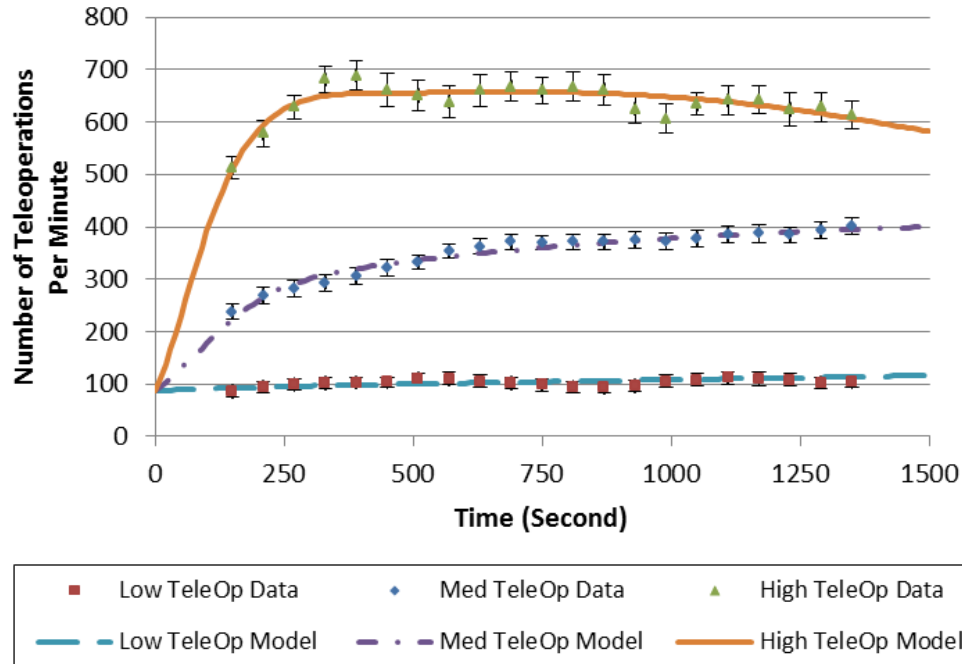
Figure 43. Number of Teleoperations: Simulation vs. Data ±1 SE

Table 10. Number of Teleoperations: Simulation to Experimental Data Fit.

Summary Statistics	Low TeleOp	Medium TeleOp	High TeleOp
Coefficient of Determination (R^2)	0.227	0.963	0.801
Root Mean Square Error (RMSE)	6.688	8.709	16.83
Root Mean Square Percent Error (RMSPE)	0.067	0.028	0.026
Mean Absolute Percent Error (MAPE)	0.055	0.022	0.020
Mean Square Error (MSE)	44.73	75.84	283.4
Bias component of MSE (U^M)	0.061	0.015	0.002
Variation component of MSE (U^S)	0.001	0.002	0.077
Covariation component of MSE (U^C)	0.938	0.983	0.921

Second, the simulation output for Found Victims is compared to average experimental data for each group in Figure 44, with the summary statistics for fit shown in Table 11. The simulations had a good fit to the experimental data with coefficient of determination (R^2) values over 0.98 for all groups. The model was able to calculate the average final number of victims found in each group within 1.2%. The only slight issue with the fit is that the performance curve for the Low TeleOp group (Figure 44a) underestimates the number of victims found for much of the earlier portion of the mission.

134

(a)

(b)

(c)
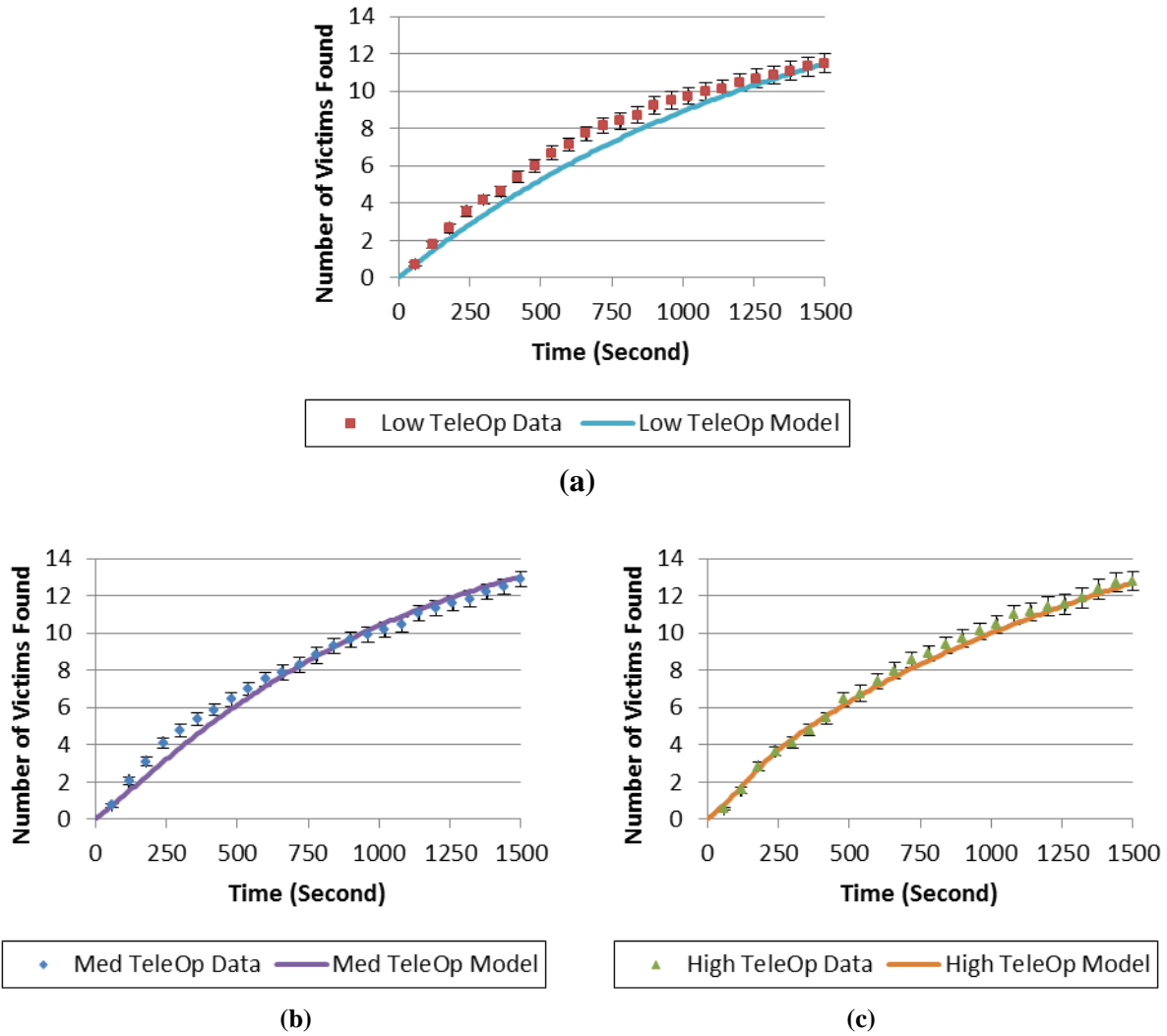
Figure 44. Found Victims: Simulation vs. Data ±1 SE: a) Low TeleOp, b) Medium TeleOp, c) High TeleOp.

Table 11. Found Victims: Simulation to Experimental Data Fit.

Summary Statistics	Low TeleOp	Medium TeleOp	High TeleOp
Percent Error at End of Mission	-0.077%	1.182%	-0.713%
Coefficient of Determination (R^2)	0.989	0.993	0.998
Root Mean Square Error (RMSE)	0.719	0.463	0.270
Root Mean Square Percent Error (RMSPE)	0.123	0.107	0.092
Mean Absolute Percent Error (MAPE)	0.102	0.068	0.048
Mean Square Error (MSE)	0.517	0.214	0.073
Bias component of MSE (U^M)	0.760	0.086	0.372
Variation component of MSE (U^S)	0.000	0.458	0.287
Covariation component of MSE (U^C)	0.240	0.457	0.342

4.2.3.4 Evaluation of Behavior and Performance Replication

The model replicated the fact that there were no significant differences in performance between the Medium and High TeleOp groups, while also capturing the differences in Number of Teleoperations of each group. Similar to Section 4.2.2., the fact that the operators in the High TeleOp group were intervening more frequently, yet did not see any additional benefit to system performance provides additional support for the hypothesis that the rate of operator intervention and the effectiveness of these interventions under high workload conditions can be in tension.

The model's predictions of Found Victims were accurate because of the interaction of two feedback loops, similar to the results seen in Section 4.2.2. First, the Trust in Automation loop provided a positive effect on Human Value Added for the higher rate of manual teleoperations of the High TeleOp group. However, the Cognitive Overload loop reduced the Human Value Added for the High TeleOp group, as the model calculated that the High TeleOp group spent almost the entire mission beyond the point of cognitive overload. This model assumption is supported by data from the experiment, where operators were asked to give a subjective assessment of their workload via the NASA-TLX scale. These NASA-TLX ratings showed that operators in the High TeleOp group reported higher temporal demand (Gao, et al., 2013), which indicates that they felt increased time pressure and that the pace of the simulation was rapid. While this measure of workload is different from the utilization metric available from the OPS-USERS testbed, NASA-TLX ratings are a commonly used and validated measure of workload (Hart, 1988).

Overall, it appears that the USARSim data supports the model assumptions and structure. The CHAS model was successfully able to replicate the behavior and performance of three groups of operators. While the accuracy of the CHAS model has been evaluated on three experimental data sets, the robustness of the model results will be evaluated in the next sections.

4.3 Sensitivity Analysis

A sensitivity analysis is an important test of the robustness of a computational model. This test asks whether the model outputs change in important ways when the assumptions are varied over the plausible range of uncertainty (Sterman, 2000). It is desirable for the outputs of a

computational model to be robust to errors in parameter estimates because it is likely that parameter estimates are imperfect. It is also helpful to identify the parameters for which the model outputs are most sensitive. This information can be useful to a system designer in two ways: a) more effort can be put into estimating the most sensitive parameters to ensure sufficient accuracy and b) it can identify the most sensitive human and system design parameters. The most sensitive human parameters represent characteristics that have a substantial impact on system performance and thus operator selection and/or training efforts should focus on these characteristics. The most sensitive system design parameters can also indicate fruitful areas for system design improvements. With limited resources for testing and development, it is desirable to determine which potential changes to a system are most likely to have strong and positive impact on system performance. Finally, regardless of how accurately model parameters are estimated, a model of human behavior and decision-making must be able to capture the inherent variability of humans. The next two sections aim to a) identify the parameters for which the model outputs are most sensitive and b) utilize Monte Carlo simulations to characterize the uncertainty in output variables due to human variability.

4.3.1 Numerical Sensitivity Analysis

The first analysis varied the values of each exogenous parameter in the original CHAS model (Figure 12) to investigate the impact on two outputs: the final area coverage performance at the end of the mission and the mean utilization throughout the mission. The baseline conditions were the parameter settings for the "All Missions" group, described in section 4.2.1 and detailed in Appendix F. The goal of this analysis was to identify the relative sensitivity of the model outputs to changes in each parameter, thus the amount of variation of each parameter needed to be consistent across parameters and large enough to perturb the model outputs. After testing different variation levels, it was determined that a 10% variation level was sufficient to perturb the model outputs. Each parameter was increased by 10% and decreased by 10% in univariate testing, where parameters were varied one at a time. The Total Number of Cells and the Sampling Interval were excluded from the analysis, as there was no uncertainty in these parameters. Also, the table function for Cognitive Overload (Section 3.4.6) was not varied in this analysis because varying a table function is a more complex process than increasing or decreasing a parameter by 10%. However, the effect of changing this table function is explored

in Section 6.1.2, where it was shown that the model was robust to changes in the point of cognitive overload as long as the model was simulating moderate workload missions.

The percent error of the output was estimated by comparing the original baseline outputs of the model to the outputs after introducing the parameter changes. The results are displayed in two charts, for the impact on area coverage performance (Figure 45) and the impact on mean utilization (Figure 46). A number of observations can be made from these results. First, the overall results show that the model was fairly robust to parameter changes, as ±10% changes in parameter values resulted in at most a 7% change in area coverage performance. Second, the results show that the most crucial relationship to estimate accurately is the non-linear relationship between Search Task Rate and Human Value Added. Base Search Task Rate is the most sensitive parameter for area coverage performance and is part of the definition of the non-linear logit relationship between Search Task Rate and Human Value Added. As described in Section 3.4.5, data was available to characterize this relationship accurately for the OPS-USERS testbed. Third, Automation Generated Search Speed is the second most sensitive parameter for area coverage performance. This parameter can be estimated accurately by collecting data on the performance of the automation without human guidance, as was done in Section 3.4.2.

For the rest of the parameters beyond the two which were most sensitive, the area coverage output changed by less than ~4% for a 10% parameter value change, indicating that model was robust to errors in parameter estimates. It should be noted that the most sensitive human parameters (defined in Section 4.1.5 as initial conditions and time constants that vary between operators), were Initial Human Trust and Time Horizon for Expected Performance. These parameters had a relatively small impact on area coverage (~3%) for a 10% change in their parameter values. However, as will be shown in Chapter 5, influencing the operator's trust in the AS and expectations of performance through various forms of priming can have a substantial impact on system performance.
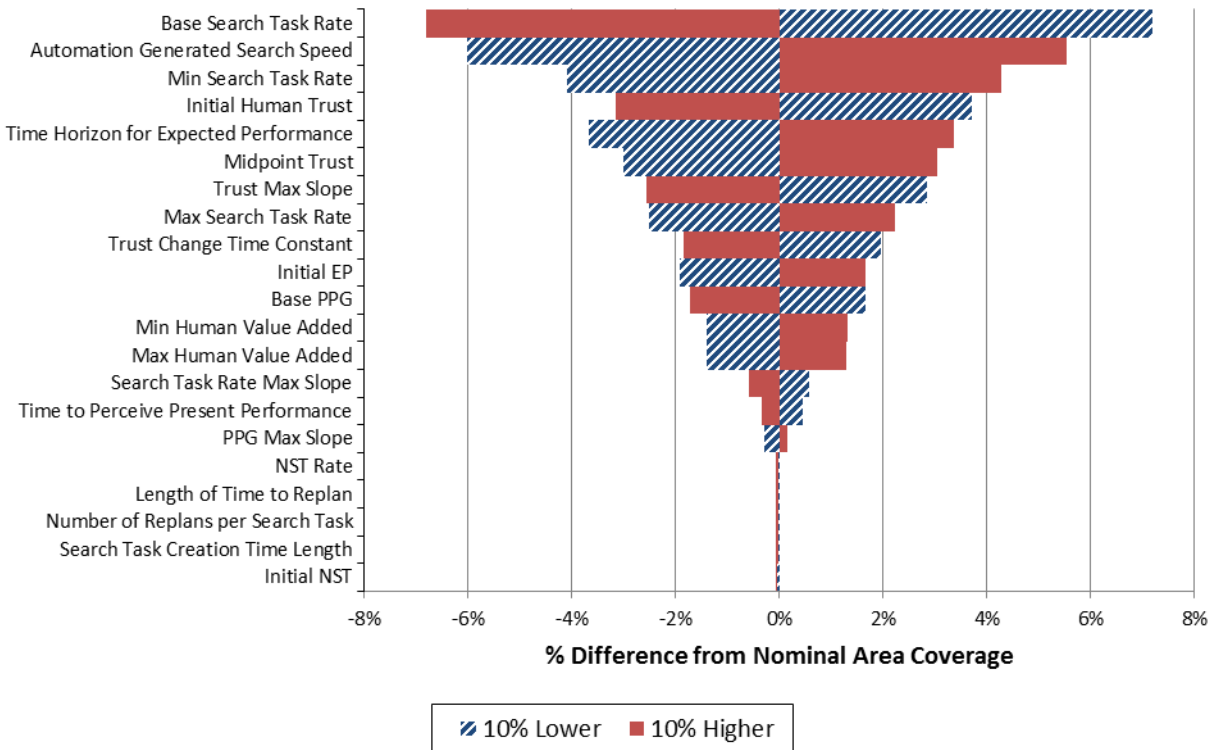
Figure 45. Impact of changes in parameter estimates on area coverage performance.
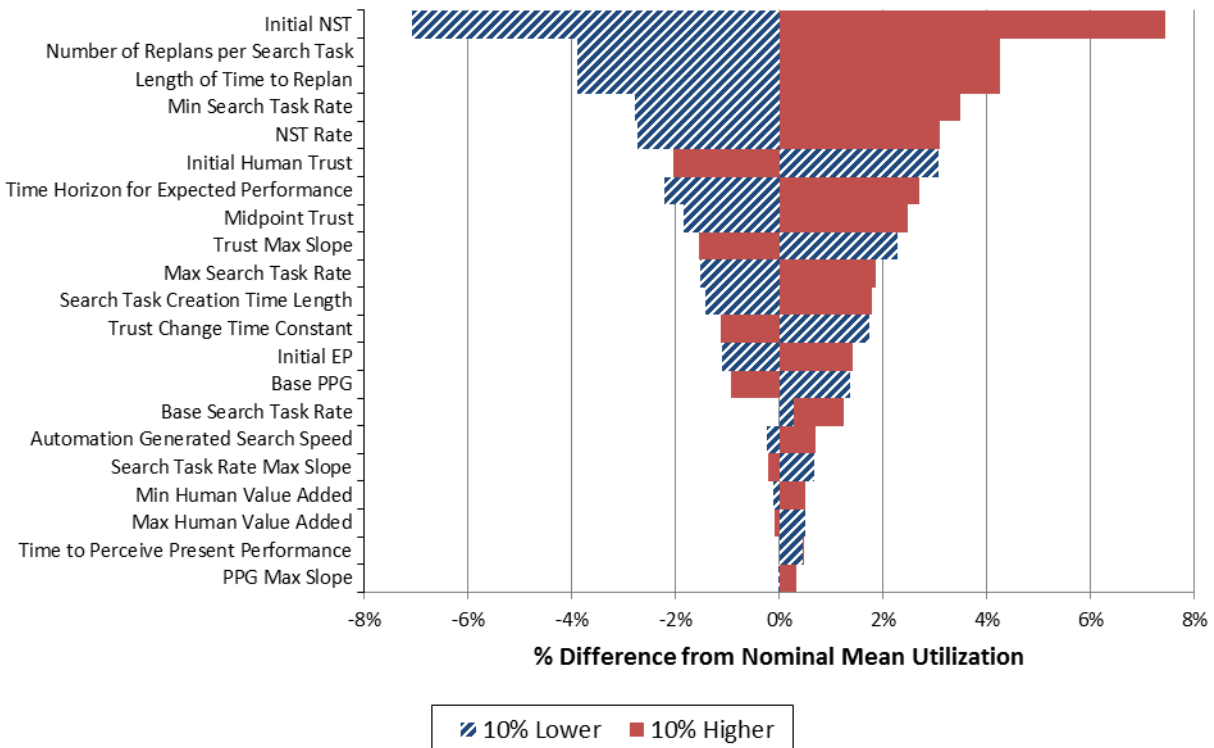
Figure 46. Impact of changes in parameter estimates on mean utilization.

Finally, a number of parameters at the bottom of Figure 45 had almost no impact on area coverage performance. All five of these parameters were inputs to the calculation of human workload. The reason that these parameters had almost no impact on area coverage performance in this analysis is that the simulations had moderate workload levels and thus the Cognitive Overload loop had little impact on area coverage. The operator's workload never exceeded 70% for these simulations and the average utilization was 41%. The experiment which the model is replicating was purposely designed to be a moderate workload experiment, thus cognitive overload is not a major concern for this data set. Experiments with higher workload where the Cognitive Overload loop was active in the simulations were described in Section 4.2.2 and 4.2.3.

However, it is still important to calculate workload accurately both to enable a system designer to analyze the impact of system changes on workload and for the model to appropriately trigger the Cognitive Overload loop. Figure 46 allows further investigation of the sensitivity of parameters to the workload calculation. The most sensitive parameter to mean utilization is Initial Nonscheduling Task load (NST), which is expected, as adding additional tasks such as identifying targets or chatting with the command center directly impacts operator workload. It should be noted that NST varies across different OPS-USERS experiments, depending on the experimental conditions, as shown in Appendix D. The OPS-USERS experimental data sets provide a good estimation of NST using the method described in Section 3.4.6. However, as previously discussed, the simple linear representation of NST in the CHAS model may not be sophisticated enough to support accurate prediction of fluctuations in workload.

Next, Number of Replans per Search Task and Length of Time to Replan were the second and third most sensitive parameters. Both of these parameters impact the rate of replan interventions and the length of these interventions, which had a weak to moderate impact on workload. All other parameters had a relatively low impact on the mean utilization (less than a 4% change in utilization for a 10% parameter change) indicating that the utilization calculation is fairly robust to errors in parameter estimates.

Overall, the model is fairly robust to changes in parameter estimates. Among the 21 parameters tested, only two (Base Search Task Rate and Automation Generated Search Speed) had a moderate impact on area coverage performance (at least a 4% change in area coverage for a 10%

140

change in parameter value). Also, only 3 parameters had a moderate impact on mean utilization, Initial NST, Length of Time to Replan, and Number of Replans per Search Task. These moderately sensitive parameters will be explored in further detail in the next section through Monte Carlo analysis.

This sensitivity analysis did not specifically identify any variables that should be removed from the model. While changes to some of the parameters did not have a large impact on area coverage performance or workload, most of these parameters are essential for defining the various non-linear relationships that were described in Chapter 3. Also, as will be discussed further in Chapter 5, modeling human time delays is crucial to the validity of the CHAS model and the ability to capture oscillatory behavior patterns, even if the impact of some of these time delays on system performance by the end of the mission is not substantial. Finally, it should be noted that a model reduction process was conducted with earlier versions of the CHAS model (Appendix A), however, the sensitivity analysis presented here used the parsimonious model presented in Chapter 3 (Figure 12).

In general, accurately estimating the automation contribution to system performance and defining relationships such as the impact of search task rate on human value added are crucial to model accuracy. Through data gathered through automation tests and human subject trials, these parameters and relationships can be estimated by the same method shown in Section 3.4.5. Given sufficient data to estimate these parameters and relationships, accurate model replications and predictions can be made.

4.3.2 Capturing Human Variability

Regardless of how accurately model parameters are estimated, a model of human behavior and decision-making must be able to capture the inherent variability of humans. Monte Carlo simulations can be utilized to generate dynamic confidence intervals for the simulation outputs. The general Monte Carlo simulation process for testing SD models (Sterman, 2000) is as follows. First, the exogenous parameters with significant uncertainty and for which the output variables are sensitive are identified. Second, probability distributions that characterize the likely values for these parameters are specified using human operator data. Then the simulation software randomly draws a value for each parameter from the chosen distribution and simulates

the model. This process is repeated for a large sample of simulations, which are used to define the confidence bounds for the output variables.

Based on the sensitivity analysis presented above, there were five candidate parameters which could be used in the Monte Carlo analysis: Base Search Task Rate, Automation Generated Search Speed, Initial NST, Length of Time to Replan, and Number of Replans per Search Task. All five of these parameters were identified as having a moderate impact on either area coverage or mean utilization (defined as greater than a 4% change in the output variable for a 10% change in the exogenous parameter value). It was decided to exclude Automation Generated Search Speed from the Monte Carlo analysis for two reasons: a) it is an estimated parameter based on model fitting (Section 3.4.2) which cannot be drawn directly from experimental data and b) this Monte Carlo analysis focused on capturing human variability as opposed to variability in the performance of the automation.

Thus, the four parameters used in the Monte Carlo analysis were: Base Search Task Rate, Initial NST, Length of Time to Replan, and Number of Replans per Search Task. All four of these variables had significant variability due to individual differences in the human operators who use the system. Using the distributions of the four variables generated from previous experimental data (Appendix K), 1000 simulations of the CHAS model were run. Once again, the baseline conditions for the model were the parameter settings for the "All Missions" group, described in section 4.2.1 and detailed in Appendix F. This enabled comparison of the Monte Carlo simulation results with the experimental data that originally informed the construction of the model (Section 3.2).

The dynamic confidence intervals generated by the Monte Carlo simulations for area coverage percentage, human workload, search task rate, and replan rate are compared to average experimental data (± 1 Standard Error (SE)) in Figure 47. Generally, the experimental data falls within the 50% confidence intervals for the Monte Carlo simulations for all four output variables. Given human variability, this is a decent measure of the accuracy of the CHAS model, along with the previously discussed evaluation of the fit of the model to historical data in Section 4.2.1.

142

Through Monte Carlo simulation, the CHAS model is able to provide a system designer with a prediction of not only the average value of system performance or workload, but also the plausible range of performance or workload that could occur. Designing systems involving human operators requires understanding not just average performance, but also the range of performance. For example, while average workload may remain below cognitive overload conditions, it is crucial to evaluate what percentage of operators may still experience cognitive overload due to human variability in using the system. System designers may want to enforce boundary conditions, such as designing the system such that 95% of operators will not go above a certain workload level. The CHAS model can aid them in this process, through simulations such as those shown in Figure 47. This type of system design to workload constraints using the CHAS model is explored further in Section 6.1.3.
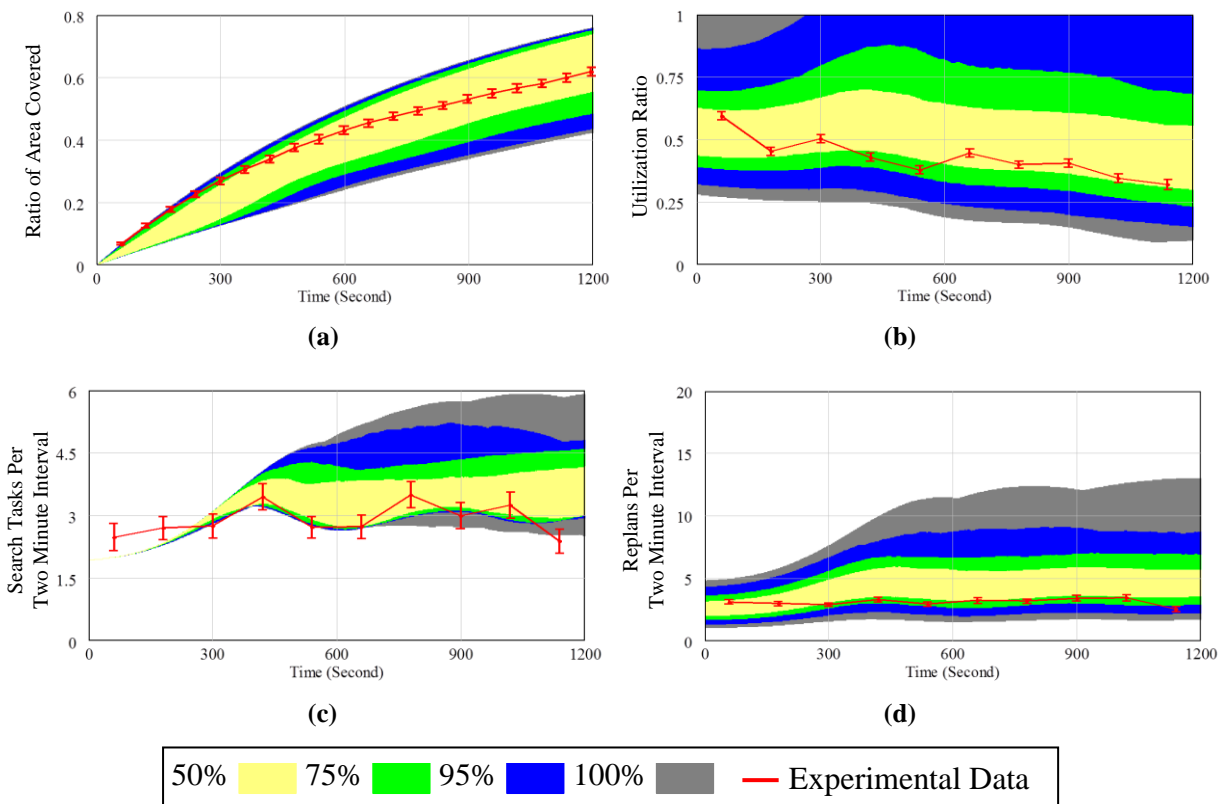
Figure 47. Dynamic confidence intervals generated via Monte Carlo simulations compared to average experimental data ± 1 SE: a) Area coverage Performance, b) Workload, c) Search Task Rate, d) Replan Rate.

There are a few limitations to this Monte Carlo approach. First, the approach assumes independence between the distributions of each parameter. For example, it is likely that an

operator with low trust in the AS will also have lower initial expectations for performance, however the simulation approach used here does not take this into account. Second, the Vensim® simulation software draws a random value from each distribution once at the start of the simulation and uses that parameter value for the entire simulation. While this is a valid assumption for initial conditions such as the Initial NST, for parameters such as the Length of Time to Replan, the model is assuming that the value drawn is an average time length that remains constant throughout the mission. This could be remedied through the use of different software or the implementation of the CHAS model in a more customizable programming language. Higher numbers of Monte Carlo simulations could also remedy this issue, although there were no substantial differences in the dynamic confidence intervals generated with different numbers of simulations (500 and 2000). Discrete Event Simulation (DES) models can more accurately re-draw a value from the distribution of Length of Time to Replan every time that a replan occurs. Additional comparisons between the CHAS model and a previously developed DES model of human supervisory control are presented in Chapter 6.

4.4 Summary

The CHAS model has been subjected to a battery of tests through the validation process presented in this Chapter. This process has built confidence in the CHAS model's accuracy and robustness, which are discussed in more detail below.

4.4.1 Model Accuracy

First, the CHAS model was able to replicate the results of the real-time human-automation collaborative scheduling experiment that provided data to inform construction of the model. The model was able to capture the differences in system performance and rates of intervention between high and low performers. The model was also able to accurately simulate changes in output variables over time, including the decline in workload throughout the mission. Additionally, the model replicated oscillations in intervention rate which were seen in the data set, as operators sought out the appropriate level of trust and the rate of intervention that would produce performance that matched their expectations.

Second, a slightly modified version of the CHAS model was able to replicate the results of a second experiment in which operators were subjected to a much higher task load. As compared to the original model presented in Chapter 3, the only change to the customized model was that the replan rate was modeled as an exogenous parameter, as operators were prompted to replan at specified time intervals. The model replicated the impact of these different replan prompting intervals accurately by showing an increase in workload, a change in search task intervention rates, and no significant difference in performance between the two groups. The model's predictions of system performance were accurate because the model simulated the impact of cognitive overload on operators who had a workload level over 70% utilization. This built confidence in the model's method of capturing the impact of cognitive overload on human decision-making and system performance. For both this data set and the original data set, the model was able to capture the major differences in workload among different groups; however, the model's replication of workload fluctuations throughout the mission was subpar, causing low goodness of fit values. This is due to the CHAS model's simple, linear representation of Nonscheduling Task Toad (NST), which cannot accurately capture fluctuations in NST seen in experimental data. Methods for enhancing the model's representation of NST are discussed in the future work section of Chapter 7.

Third, as an external validation, a tailored version of the CHAS model was also used to replicate a data set from a multi-robot Urban Search and Rescue (USAR) experiment. The only major change to model structure was a simplification of the workload calculation. The model accurately captured the impact of an increased rate of manual teleoperation on system performance. The learning behavior of operators in the data set was accurately replicated. Once again, the model captured the diminishing returns of extremely high rates of teleoperation due to cognitive overload. This external validation test demonstrated the ability to generalize the model for use with other real-time human-automation collaborative scheduling systems.

4.4.2 Model Robustness

In addition to testing the model's ability to accurately replicate experimental data, the validation tests presented in this chapter also evaluated the model's robustness under a variety of conditions.

First, the adequacy of the model boundary was evaluated by comparing the endogenous and exogenous variables in the model to determine whether the model is appropriate for the purpose for which it was built and whether the model includes all relevant structure. Also, tests were conducted to show that the model structure captures important aspects of real-time human-automation collaborative scheduling, such as the tension between the positive impact of operator interventions and the effectiveness of those interventions once operator cognitive workload reaches too high of a level.

Second, an extreme conditions test was conducted by increasing the Automation Generated Search Speed by 1200%. The model behaved as expected by quickly reaching, but not exceeding 100% area coverage performance. Additional extreme condition testing showed that through the use of logit curves to characterize key causal relationships, the model is robust to extreme conditions such as large differences between expected and perceived performance. In addition, an integration error tested demonstrated that the model was robust to changes in the time step and integration method used for simulation.

Third, a numerical sensitivity analysis was conducted to evaluate whether model outputs change in important ways when there are errors in parameter estimates. The analysis demonstrated that the model is not overly sensitive to errors in parameter values, but did indicate that accurately estimating certain relationships, such as the impact of search tasks on system performance, is crucial to model accuracy. Through data gathered through human subject trials, sufficient data can be gathered to estimate these relationships, enabling accurate model replications and predictions.

Finally, through Monte Carlo simulations, the CHAS model was able to characterize the impact of human variability on system performance. The CHAS model is able to provide a system designer with a prediction of not only the average value of system performance or workload, but also the plausible range of performance or workload that could occur. This is beneficial to a system designer who wants to evaluate the impact of system changes on the boundary conditions of system design.

Having built confidence in the CHAS model's replication accuracy and robustness, Chapter 5 will describe an additional human subject experiment that was run a) to gather additional data to

146

evaluate model assumptions and b) to test the ability of the CHAS model to predict the impact of system changes on system performance.

5 Predictive Validation Experiment

This chapter describes a human subject experiment that was conducted to evaluate the ability of the Collaborative Human-Automation Scheduling (CHAS) model to predict the impact of changes in system design and operator training on human and system performance. First, the CHAS model is used to develop experimental hypotheses, including quantitative predictions of performance. The test subjects, apparatus, experimental design, and procedure are presented. The experimental results are presented, including an analysis of the impact of demographics on performance. Data is presented to evaluate the assumptions built into the CHAS model. Finally, the experiment results are compared to the predictions made by the CHAS model.

5.1 Experimental Objectives

The primary objective of this experiment was to validate predictions from the CHAS model. The secondary purpose was to test the impact of design changes in the Onboard Planning System for UVs in Support of Expeditionary Reconnaissance and Surveillance (OPS-USERS) testbed. The tertiary purpose was to investigate how human trust in an Automated Scheduler (AS) changes throughout a simulated mission controlling a team of UVs. This experiment was designed to test the dynamic hypothesis developed through data analysis and development of the CHAS model. As described in Section 3.3, the dynamic hypothesis of the CHAS model is that if operators can either a) anchor to the appropriate trust in the AS and expectations of performance earlier in the mission and/or b) adjust their trust and expectations faster through better feedback about the AS, then system performance should improve.

5.2 Experimental Hypotheses

Following the multi-stage validation process described in Chapter 4, the CHAS model was used to develop hypotheses of how system design and training changes could potentially improve system performance. Commonly referred to as "policy implications" testing in the SD modeling community (Sterman, 2000), the model was exercised in a number of simulation runs to identify changes to a baseline system that could improve the performance of the system. While these policies could include both changing exogenous parameters and changing the feedback structure of a system, this analysis focused only on modulating exogenous parameters.

Thus, the goal was to make quantitative predictions of the impact of changing certain parameters on the primary performance metric of area coverage by the end of the mission. The baseline conditions for the model were the parameter settings, described in section 4.2.1 and detailed in Appendix F, which replicated the average behavior of all test subjects in a previous experiment using the OPS-USERS testbed.

The three exogenous parameters selected for testing were: Initial Human Trust, Initial Expectations of Performance (EP), and Time to Perceive Present Performance (TPPP). The first two parameters were chosen because the sensitivity analysis described in section 4.3.1 identified Human Trust and Expectations of Performance (EP) as the most sensitive human attributes. These parameters represent characteristics that have a substantial impact on system performance and thus operator selection and/or training efforts should focus on these characteristics. The third parameter, TPPP, was chosen because previous research has shown that information time delays can have serious consequences for human decision-making in dynamic systems (Brehmer, 1990; Sterman, 1989b). While the manipulation of other variables, such as the Automation Generated Search Speed, may have had a more substantial effect on performance (Section 4.3.1), this research focused on modeling the human operator and understanding how humans collaborate with an AS to improve system performance. Thus, all three independent variables focused on training methods and system design changes to influence operator behavior, both to assess the impact on system performance and to evaluate the assumptions and accuracy of the CHAS model.

The following sections describe the changes to training procedures and system design that were implemented in the testbed to attempt to modulate these variables. Also, CHAS model predictions for the impact of changing these three parameters on human and system performance are presented.

5.2.1   Initial Trust Level

According to the CHAS model, the Initial Human Trust of the operator should have an impact on the performance of the system. While the sensitivity analysis in Section 4.3.1 showed that Initial Human Trust had a weak to moderate impact on area coverage performance, a strong enough change in Initial Human Trust should result in a detectable change in system performance. Thus,

it was decided to prime test subjects prior to their interaction with the AS in order to influence their initial trust level. Priming has been studied extensively in the psychology and neuroscience domains (Cave, 1997; Henson, 2003; Kosslyn & Rosenberg, 2011; Schacter, 1987; Schacter & Buckner, 1998) and it is known that humans are susceptible to anchoring biases in decision-making and judgments under uncertainty (Dzindolet et al., 2002; Tversky & Kahneman, 1974). However, there has been little research on the impact of priming on operators controlling multiple UVs.

A few studies have investigated the impact of framing on human decision-making and reliance on an automated decision aid (Dzindolet, et al., 2002; Lacson, Wiegmann, & Madhavan, 2005). In these experiments, test subjects were provided with information about previous automation performance with either positive framing ("the aid usually made about half as many errors as most participants") or negative framing ("the aid usually made about 10 errors in 200 trials"). These studies found that the manner in which information about the reliability of the automation was presented to operators could subtly influence reliance on the automation, but all experiments focused on signal detection, rather than the more complex decision-making required for controlling multiple UVs. In another study, Rice et. al (2008) primed test subjects with images of automation with either positive and negative affect. They found that operators primed with positive images had faster reaction times and higher accuracy in a visual identification task with the assistance of an automated identification aid. However, this was the only task that the operators were conducting, as opposed to the testbed described in this thesis where operators were multi-tasking. Also, the automation was for target identification and had 100% reliability, as opposed to the automated scheduling algorithm used in this testbed which has been found to be provably good, but suboptimal (Choi, et al., 2009; Whitten, 2010).

Thus, the first independent variable for this human subject experiment was called "*A Priori* Priming," with three levels: "Positive Priming," "Negative Priming," and "No Priming." After completing a self-paced, slide-based tutorial about the testbed, operators read a passage containing six actual quotes written by test subjects from a previous experiment using this testbed. For the Positive Priming level, the quotes were all from operators who had positive impressions of the AS, describing it as fast, intuitive, easy to use, and smart. For the Negative Priming level, the quotes were all from operators who were dissatisfied with the AS, describing

how they did not agree with the plans made by the AS, how they wished they could manually assign tasks, and how the AS made poor decisions. Both passages are shown in Appendix K. The No Priming level served as a control group, where operators did not receive a passage to read after training. This was a between-subjects factor, in that a particular subject only experienced one Priming Level, to avoid training biases and confusion.

In order to provide quantitative predictions of the impact of positive and negative priming, data from the twelve operators who wrote the quotes used in the priming passages was analyzed. All test subjects in the previous experiment were asked to rate their satisfaction with the plans created by the AS after each mission on a Likert scale from 1-5 (low to high), as explained in Section 3.2. This subjective rating was used as a proxy variable for trust in the AS. On average across all operators, the average rating of satisfaction was 2.82 out of 5. Among the six operators with positive quotes, the average rating was 3.16, a 12% increase. Among the six operators with negative quotes, the average rating was 2.16, a 23% decrease.

For the purposes of making quantitative predictions, it was assumed that positive priming would cause a 12% increase from the baseline Initial Human Trust, while negative priming would cause a 23% decrease from the baseline. No Priming was the control group, and thus the baseline Initial Human Trust was maintained. The results of these model predictions are shown in Table 12.

Table 12. Model predictions for impact of *a priori* priming on area coverage performance.

A Priori Priming Level	% Change in Initial Human Trust	Area Coverage Performance
Negative Priming	-23%	68.6%
No Priming	0%	62.7%
Positive Priming	+12%	60.3%

These predictions were based on the assumption that priming would change the average initial trust of each group by the desired amounts. It should be noted that the model predicted that a decrease in trust would improve performance while an increase in trust would lower system performance. Once again, the model made these predictions because the automated scheduling algorithm used in this testbed has been found to be provably good, but suboptimal (Choi, et al., 2009; Whitten, 2010). Thus, lowering trust should lead operators to intervene more frequently in

152

order to guide the suboptimal automation. Based on these predictions, the following results were expected:

- *Hypothesis 1*: Negative *a priori* priming of human trust in the AS is expected to result in a 9% increase in system performance by the end of the mission, as compared to the no priming control condition.

- *Hypothesis 2*: Positive *a priori* priming of human trust in the AS is expected to result in a 4% decrease in system performance by the end of the mission, as compared to the no priming control condition.

- *Hypothesis 3*: Negative *a priori* priming of human trust in the AS is expected to result in a 12% increase in system performance by the end of the mission, as compared to the positive priming condition.

5.2.2   Expectations of Performance

In addition to Initial Trust, the operator's expectations of performance should have an impact on system performance, according to the CHAS model. Once again, while the sensitivity analysis in Section 4.3.1 showed that Initial Expected Performance (EP) had a weak to moderate impact on area coverage performance, a strong enough change in operator expectations of performance should result in a detectable change in system performance. In order to influence operators' expectations of performance, a new performance plot was implemented in the testbed, shown in Figure 48. The old performance plot (Section 3.2.1) showed scores for the current and proposed schedule based on the priority levels of assigned tasks in each schedule. The new performance plot showed a reference area coverage "expectation" line in red along with the actual area coverage performance thus far in the mission in blue. During training, test subjects were told that the red line was the average of area coverage accomplished by previous users of the system. This reference line was meant to serve as a different form of priming, by setting operators' expectations of how well the system should perform based on the how well other operators had performed before them.

The second independent variable in the experiment was called "Real-Time Priming," with two levels: "Low Priming" and "High Priming." The Low Priming level showed the average area coverage curve of the low performers group from the previous experiment (Section 3.2) as the red reference line, with the final area covered at the end of mission equal to 53.2% (Figure 48a). The High Priming level showed the average area coverage curve of the high performers group, with the final area covered equal to 76.6% (Figure 48b). This was also a between-subjects variable, in that a particular subject only experienced one Real-Time Priming Level, to avoid confusion. It was possible for a test subject to perceive that they were doing "better than expected" (Figure 48a) or "worse than expected" (Figure 48b) through the performance plot.
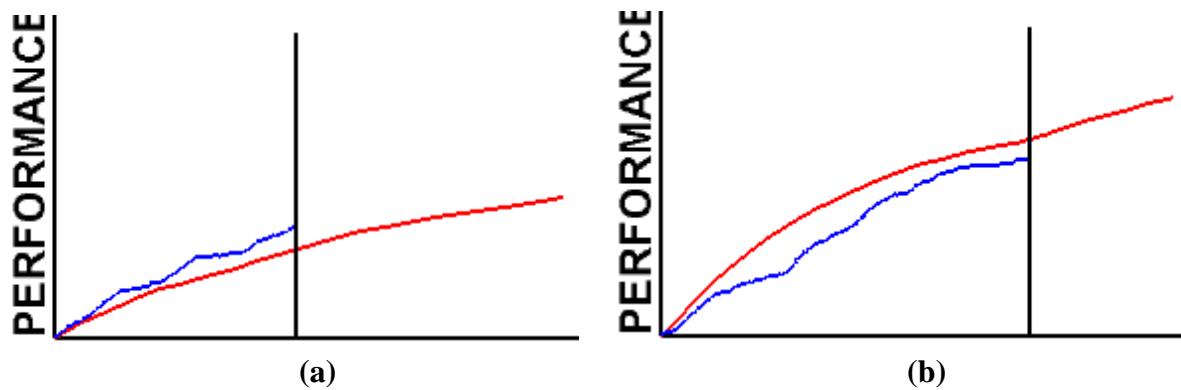
Figure 48. New performance plot: (a) Low reference line. (b) High reference line.

In order to provide quantitative predictions of the impact of real-time priming, the Initial EP parameter values of the low and high performer groups from the previous experiment were compared. When the model was fit to data from the low performer group, the Initial EP parameter value was 4.00 cells/second (Section 4.2.1 and Appendix F). In contrast, when the model was fit to data from the high performer group, the Initial EP parameter value was 6.25 cells/second, a 56% increase. Thus, for the purposes of making quantitative predictions, it was assumed that High Real-Time Priming would cause a 56% increase from the baseline Initial EP. All other parameter values were maintained at the baseline level. The results of these model predictions are shown in Table 13.

Table 13. Model predictions for impact of real-time priming on area coverage performance.

Real-Time Priming Level	% Change in Initial EP	Area Coverage Performance
Low Priming	0%	62.7%
High Priming	+56%	67.5%

154

These predictions were based on the assumption that High Real-Time Priming would change the system performance expectations by the desired amount, which would cause operators in the High Real-Time Priming group to have a larger Perceived Performance Gap (PPG). It was also assumed that Low Real-Time Priming would not substantially change performance expectations as compared to the average operator. Based on these predictions, the following results were expected:

- *Hypothesis 4*: High real-time priming of operator expectations of performance is expected to result in an 8% increase in system performance by the end of the mission as compared to the low real-time priming condition.

5.2.3   Time to Perceive Present Performance

A third parameter investigated was the Time to Perceive Present Performance (TPPP). TPPP is an assumption about how long it takes the operator to detect changes in the area coverage rate. Given that the testbed now provided a plot showing real-time updates of area coverage performance throughout the mission, TPPP could potentially be increased by delaying the reporting of performance in the plot, as shown in Figure 49. While there was no way for the experimental testbed to gather precise data on human visual perception of information, operator actions and system performance could be measured to evaluate the impact of this system design change.
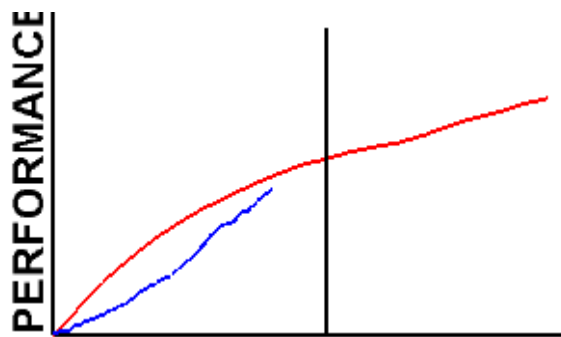
Figure 49. Performance plot showing time delay of performance feedback.

The impact of time lags on the manual teleoperation of robots has been studied extensively (Chen, Haas, & Barnes, 2007; Ricard, 1994; Sheridan, 1993; Thompson, Ottensmeyer, & Sheridan, 1999; Wickens & Hollands, 2000), but information time delays are less frequently

155

studied in human supervisory control because the operator does not have direct control over the vehicle and thus there is no delay in the control loop that can lead to instability (Sheridan, 1992). Walker et al. (2012) investigated the impact of latencies and predictive displays on supervisory control of a swarm of robots, where all information, including the locations of the robots, was delayed. Southern (2010) examined the impact of communication delays among a human supervised multi-UV system on performance, based on the testbed described in this thesis. However, the experiment described here attempted to evaluate the impact of an information reporting time delay of system performance on human decision-making, while there were no communications delays among the team of UVs.

Thus, the third independent variable was called "Information Time Delay," with two levels: "No Delay" and "With Delay." For the No Delay level, the performance plot was just as shown in Figure 48b, where the area coverage performance blue line was updated immediately. For the With Delay level, a 120-second information reporting delay was implemented, as shown in Figure 49, with the blue line filled in after a 120-second delay. A 120-second delay was chosen because the model sensitivity analysis in Section 4.3.1 showed that ±10% changes (±30 seconds) in TPPP did not have a strong enough impact on system performance. Thus, the delay was raised to 120-seconds and pilot testing confirmed that operators noticed the information reporting delay. It should be noted that this was a within-subjects factor, in that each test subject experienced both levels of Information Time Delay.

In order to provide quantitative predictions of the impact of the information time delay, TPPP values for operators in each Information Time Delay group had to be estimated. The TPPP parameter value was set to 10 seconds for the No Delay group, assuming that there was still an inherent perception delay due to operator attention allocation inefficiencies (Cummings & Mitchell, 2008). TPPP was set to 130 for the With Delay group, adding in the 120-second delay. All other parameter values were maintained at the baseline level. The results of these model predictions are shown in Table 14.

Table 14. Model predictions for impact of information time delay on area coverage performance.

Information Time Delay Level	TPPP parameter value	Area Coverage Performance
No Delay	10	66.1%
With Delay	130	63.1%

These predictions were based on the assumption that a time lag in the performance plot would in fact delay the perception of performance of each operator. Once again, while there was no way for the experimental testbed to measure the actual time delay in operator perception of changes in system performance, operator actions and system performance could be measured to evaluate the impact of this system design change. It should be noted that the predicted difference in system performance was small (5%) and thus it was likely that the experimental data would not have statistically significant differences. However, based on these predictions, the following results were expected:

- *Hypothesis 5*: the addition of an information reporting time delay to the performance plot is expected to result in a 5% decrease in system performance by the end of the mission.

5.3 Test Subjects

To test these hypotheses, 48 test subjects were recruited from undergraduate students, graduate students, and researchers at the Massachusetts Institute of Technology (MIT). As the concept of multiple UV supervisory control through a decentralized network is a futuristic concept, without current subject matter experts, it was determined that a general user base would provide a sufficient sample of potential future UV system operators.

The 48 test subjects consisted of 35 men and 13 women. The age range of test subjects was 18-32 years with an average age of 23.08 and a standard deviation of 3.84. Only 5 test subjects had served or were currently serving in the military, but a previous experiment using the OPS-USERS testbed showed that there was no difference in performance or workload between test subjects based on military experience (Cummings, Clare, et al., 2010). Each participant filled out a demographic survey prior to the experiment that included age, gender, occupation, military experience, average hours of television viewing, video gaming experience, and perception of UAVs. Additionally, all test subjects filled out a 52-question Metacognitive Awareness Inventory (MAI) prior to conducting the experiment. Metacognitive awareness "refers to the ability to reflect upon, understand, and control one's learning" (Schraw & Dennison, 1994, p. 460) which is relevant to expectation setting and adjustment. The consent forms and demographic surveys filled out by test subjects can be found in Appendices L and M. Descriptive statistics of the results of these demographic surveys can be found in Appendix N.

5.4 Apparatus

The human subject experiment was conducted using two Dell 17" flat panel monitors operated at 1280 x 1024 pixels and a 32-bit color resolution. The primary monitor displayed the testbed and the secondary monitor showed a legend of the symbols used in the system (Appendix O). The workstation was a Dell Dimension DM051 with an Intel Pentium D 2.80 GHz processor and a NVIDIA GeForce 7300 LE graphics card. System audio was provided using standard headphones that were worn by each participant during the experiment. All data regarding the test subjects' interactions with the system for controlling the simulated UVs was recorded automatically by the system.

5.5 Experimental Design

Three scenarios were designed for this experiment: a practice scenario and two test scenarios. Each scenario involved controlling four simulated UVs (one of which was weaponized) in a mission to conduct surveillance of an area in order to search for targets, track these targets, and destroy any hostile targets found (when instructed). The area contained both water and land environments and targets could be either tanks on the ground or boats in the water. The vehicles automatically returned to the base when necessary to refuel and were equipped with sensors (either radar or cameras) which would notify the operator when a target was detected so that the operator could view sensor information in order to designate the target and give it a priority level. Perfect sensor operation was assumed, in that there were no false detections or missed target detections.

Each scenario had 10 targets that were initially hidden to the operator. These targets always had a positive velocity and moved on pre-planned paths throughout the environment (unknown to the operator), at roughly 5% of the cruise velocity of the WUAV. Each scenario had three friendly targets, three hostile targets, and four unknown targets. The operator received intelligence information on the unknown targets through the chat window, revealing that two of the targets were friendly and two were hostile. Upon receiving this intelligence, the operator could re-designate the targets. The operator would also be asked by the "Command Center" through the chat window to create search tasks in specified quadrants at various times throughout the mission. Finally, new Rules of Engagement (ROEs) were presented to the operator through the

158

chat window every 5 minutes during the 20 minute mission. The ROEs instructed operators on aspects of the mission that were most important at the time in order to guide their high level decision making. The ROEs also specified when hostile target destruction was permitted. The ROEs are listed in Appendix P. The scenarios were all different, but of comparable difficulty, so that operators would not learn the locations of targets between missions.

5.5.1 Independent Variables

The experimental design was a 3x2x2 repeated measures design with three independent variables: 1) the *A Priori* Priming level based on a passage that was read by operators immediately following training, 2) the Real-time Priming level of the reference area coverage curve shown in the performance plot, and 3) the Information Time Delay of feedback on actual area coverage performance through the plot. As described above, the *A Priori* Priming variable had three levels: Positive Priming, Negative Priming, and a No Priming control condition. The Real-Time Priming variable had two levels: Low Priming and High Priming. Finally, the Information Time Delay variable had two levels: No Delay and With Delay. Information Time Delay was a within-subjects factor, as each subject experienced both a No Delay and With Delay mission. These missions were presented in a randomized and counterbalanced order to avoid learning effects.

5.5.2 Dependent Variables

The dependent variables for the experiment were mission performance, primary workload, secondary workload, Situation Awareness (SA), and subjective ratings, taken both during the missions and post-mission. Overall mission performance was measured by taking the following five metrics: percentage of area coverage, percentage of targets found, percentage of time that targets were tracked, number of correct hostile targets destroyed, and number of mistaken targets destroyed. The primary workload measure was a utilization metric calculating the ratio of the total operator "busy time" to the total mission time. For utilization, operators were considered "busy" when performing one or more of the following tasks: creating search tasks, replanning, identifying and designating targets, approving weapons launches, interacting via the chat box, and answering a survey question. All interface interactions were via a mouse with the exception of the chat messages, which required keyboard input.

An alternative method for assessing workload was measuring the spare mental capacity of the operator through reaction times to a secondary task. Secondary workload was measured via reaction times to text message information queries, as well as reaction times when instructed to create search tasks via the chat tool. Such embedded secondary tools have been previously shown to be effective indicators of workload (Cummings & Guerlain, 2004).

SA was measured through the accuracy percentage of responses to periodic chat box messages querying the participant about aspects of the mission. Additionally, four of the targets were originally designated as unknown. Chat messages provided intelligence information to the operator about whether these targets were actually hostile or friendly (based on their location on the map). It was up to the operator to re-designate these targets based on this information. Therefore, a second measure of SA was the ratio of correct re-designations of unknown targets to the number of unknown targets found.

Throughout the mission, a pop-up survey window (Figure 50) appeared in the lower left corner of the Map Display to ask the operator to provide three ratings. Operators read the following during their training session about the survey window:

Every 2 minutes throughout all of your missions, this window will appear in the bottom left corner of the Map Display. It asks you to rate three questions:

- *Rate how well you think the system is performing right at this moment (1=Extremely Poor, 4=OK, 7=Extremely Well)*
- *Rate how well you expect the system to be performing at this moment (1=Extremely Poor, 4=OK, 7=Extremely Well)*
- *Rate your trust in the Automated Scheduler that you work with in the SCT (1=No Trust, 4=Neutral, 7=Absolute Trust)*

Figure 50. Pop-up survey window.

160

This survey was used to gather near real-time data on the operator's perception of performance, expectations of how well the system should be performing, and trust in the AS. All three of these variables were essential to the CHAS model and collecting this data enabled the evaluation of model assumptions. A fourth metric, the Perceived Performance Gap (PPG) was calculated by taking the percent difference between the expectation and performance rating during data analysis. A Likert rating scale of 1-7 (low to high) was used simply because that is the scale that is used in an empirically validated and commonly used trust in automation questionnaire (Jian, Bisantz, & Drury, 2000). These questions were asked every 2 minutes, starting at 60 seconds into the mission. The goal was to sample the operator's perceptions, expectations, and trust level as frequently as possible without distracting the operator from his or her primary tasks. Based on the fact that operators typically replan (use the AS to generate a new schedule) 1-2 times per minute, it was decided that prompting the user every minute would be too frequent and distracting. Instead, a questioning interval of 2 minutes was adopted. Pilot tests verified that this frequency of sampling captures the dynamics of perceptions, expectations, and trust without being too intrusive. Online probes to gather subjective ratings are commonly used in experiments such as these to measure workload and SA (Endsley, Sollenberger, & Stein, 2000), and they have been proposed as a method to measure trust (Miller & Perkins, 2010).

A survey was provided at the end of each mission asking the participant for a subjective rating of their confidence, workload, and satisfaction with the plans generated by the AS on a Likert scale from 1-5 (low to high). At the end of the entire experiment, test subjects filled out a 12-question survey which is commonly used to measure trust in automation and has been empirically validated (Jian, et al., 2000). All of these subjective ratings are crucial, both for providing an additional measure of workload and for evaluating whether the independent variables influenced the operator's confidence and trust in the collaborative decision-making process, factors which have been shown to influence system performance (Parasuraman & Riley, 1997).

5.6 Procedure

In order to familiarize each subject with the interface, a self-paced, slide-based tutorial was provided (Appendix Q). Subjects then conducted a fifteen-minute practice session during which the experimenter walked the subject through all the necessary functions to use the interface. Each

subject was given the opportunity to ask the experimenter questions regarding the interface and mission during the tutorial and practice session. Each subject also had to pass a proficiency test, which was a 5-question slide-based test (Appendix R). If the subjects did not pass the proficiency test, they were given time to review the tutorial, after which they could take a second, different proficiency test. All subjects passed on the first test.

The actual experiment for each subject consisted of two twenty-minute sessions, one for each of the two Information Time Delay levels. The order of the Information Time Delay levels presented to the subject was counterbalanced and randomized to prevent learning effects. During testing, the subject was not able to ask the experimenter questions about the interface and mission. All data and operator actions were recorded by the interface and Camtasia$^©$ was used to record the operator's actions on the screen. Finally, a survey was administered at the end of each mission to obtain the participant's subjective evaluation of their workload, confidence, and trust, along with general comments on using the system (Appendix S). Subjects were paid $10/hour for the experiment and a performance bonus of a $100 gift card was given to the individual who obtained the highest mission performance metrics (to encourage maximum effort).

5.7 Results

This section discusses the results of the experiment and compares them to the hypotheses described in Section 5.2. The section begins with an analysis of the value that human operators were adding to system performance as compared to the automation-generated performance. Then, performance, workload, SA, and subjective response results are compared across the three independent variables. Demographic predictors of performance are discussed and qualitative survey comments are presented to provide insight on operator strategy and trust. A more detailed analysis of the impact of gaming frequency on the reaction to the independent variables and system performance is presented. Finally, some of the assumptions in the CHAS model which were previously untested were evaluated using data gathered for the first time in this experiment.

5.7.1 Human Value Added

A histogram of the area coverage performance by the end of the mission of all 48 subjects is shown in Figure 51. Also shown in the diagram is a red line indicating the area coverage

performance of the "obedient human" mission, which was described in Section 3.4.2. The percent area covered in the obedient human mission was 56.2%, while the average human operator in this experiment achieved area coverage of 63.0%, a 12% increase in performance due to human value added. Eighty of the 96 missions (83%) had improved performance as compared to the obedient human mission. The top performer in the experiment achieved area coverage of 85.0%, a 51% increase in performance over the obedient human condition.
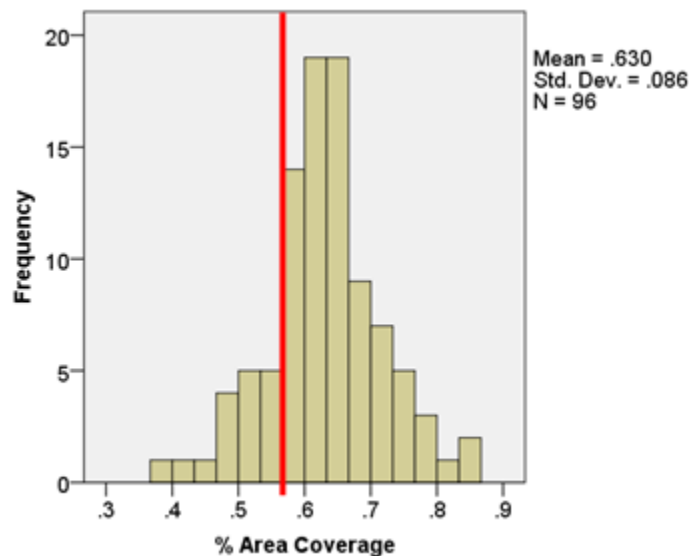
Figure 51. Test subject area coverage performance compared to "obedient human" mission (red line).

This aligns with previous results using this testbed and adds additional support to the assumption that the human operator can add value over the automation generated performance. It should also be noted that 16 of the missions (17%) had lower area coverage performance than the obedient human mission, and thus modeling the potential for negative human value added is also an important facet of the CHAS model.

5.7.2   Impact of Independent Variables

A statistical analysis of all dependent variables was conducted and details, including descriptive statistics of all measures and testing for order effects, are provided in Appendix T. The following sections discuss the results for each independent variable.

5.7.2.1 *A Priori* Priming

Beginning with the *A Priori* Priming independent variable, results showed that test subjects who experienced the Positive Priming level had higher ratings of trust in the AS following their training mission, prior to the actual experimental missions. Pairwise Mann-Whitney comparisons showed that the Positive Priming group had 13.8% higher trust ratings compared to the No Priming control group ($Z = -2.570$, $p = 0.010$) and 20.7% higher trust ratings compared to the Negative Priming group ($Z = -3.186$, $p = 0.002$). There were no significant differences in pre-experiment trust ratings between the Negative Priming group and the No Priming control group ($Z = -0.807$, $p = 0.420$). Similar results were found for the average real-time rating of trust during the missions, where the Positive Priming group had 14.9% higher trust ratings compared to the No Priming control group ($Z = -2.614$, $p = 0.009$) and 24.2% higher trust ratings compared to the Negative Priming group ($Z = -3.741$, $p < 0.001$). Again, there were no significant differences in average real-time trust ratings between the Negative Priming group and the No Priming control group ($Z = -1.036$, $p = 0.300$). These results are shown in Figure 52.
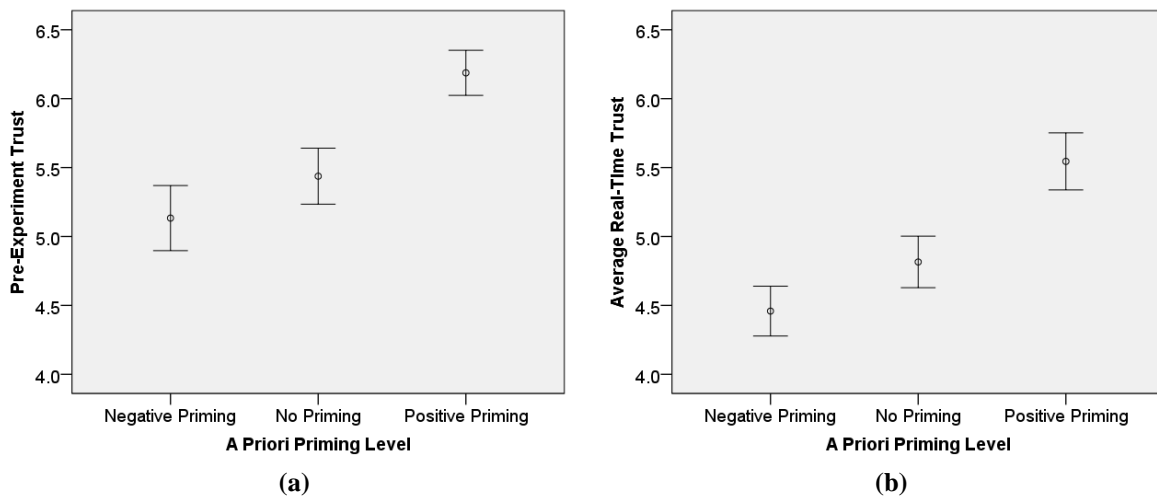
Figure 52. Trust ratings comparison: (a) Pre-experiment ratings. (b) Average real-time ratings during the mission. Standard Error bars are shown.

It appears that positive *a priori* priming had the desired effect of raising initial trust in the automation. This higher trust level among the positive priming group was maintained on average throughout the experiment, although trust during the mission was lower than pre-experiment trust, as shown in Figure 52b. It should be noted that the impact of negative priming was not strong enough to significantly lower self-reported trust as compared to the control group. This

may be due to the particular method of priming that was used in this experiment, such as the quotes that were chosen (Appendix K). It could also show that operators were willing to raise their trust level in automation based on information about how it has worked in the past, but lowering their trust level requires actually working with the automation in person.

Overall, these results show that priming can be effective at influencing initial trust level, especially when the priming is meant to raise trust. Interestingly, after the end of the experiment, there were no significant differences in trust across the three *A Priori* Priming groups according to a Kruskal-Wallis omnibus test of the 12-question post-experiment trust survey data ($\chi^2(2, N=48) = 1.986$, $p = 0.371$). Thus, this data also provides evidence that operators adjust their trust level over time as they work with the AS and thus the effects of priming are not enduring. This aligns with previous empirical evidence demonstrating that trust has inertia (Lee & Moray, 1994; Lewandowsky, et al., 2000; Parasuraman, 1993), where the effect of perceived automation performance on trust is not instantaneous, but trust does change steadily over time.

In terms of system performance, the only significant difference among the three *A Priori* Priming groups was in terms of target destruction mistakes. These mistakes could take two forms: a) destroying a target against the ROEs or b) incorrectly destroying a friendly target. There were 14 missions with a mistaken target destruction and 11 of these missions were conducted by operators who experienced Negative *A Priori* Priming (Figure 53). This difference was statistically significant according to a Kruskal-Wallis omnibus test, $\chi^2 (2, N=14) = 13.0$, p=0.002.
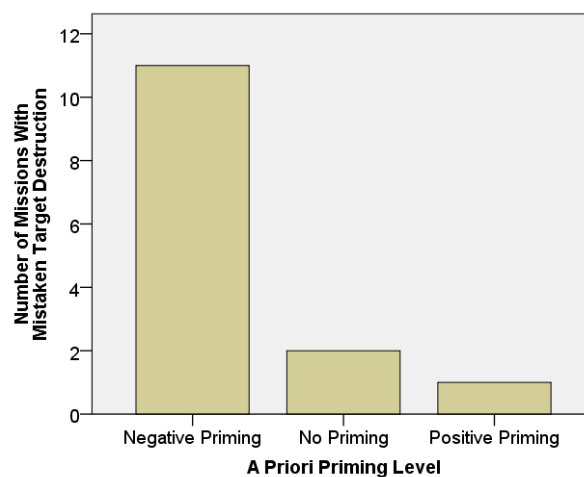
Figure 53. Number of missions with mistaken target destruction across *A Priori* Priming levels.

Notably, there were no other significant differences in the system performance metrics, including the primary metric of area coverage, across the *A Priori* Priming groups. However, real-time ratings indicated that operators in the Negative Priming group recognized their mistakes. For the average real-time rating of how well operators thought the system was performing, pairwise Mann-Whitney comparisons showed the Negative Priming group had 7.0% lower self-performance ratings compared to the No Priming control group ($Z = -1.832$, $p = 0.067$) and 9.9% lower performance ratings compared to the Positive Priming group ($Z = -2.160$, $p = 0.031$). There were no significant differences in performance ratings between the Positive Priming group and the No Priming control group ($Z = -0.833$, $p = 0.405$). Similar results were found for the average real-time rating of how well operators expected the system to perform, where the Positive Priming group had 10.4% higher expectations ratings compared to the No Priming control group ($Z = -2.674$, $p = 0.007$) and 13.1% higher expectations ratings compared to the Negative Priming group ($Z = -3.189$, $p = 0.001$). There were no significant differences in average real-time expectations ratings between the Negative Priming group and the No Priming control group ($Z = -0.395$, $p = 0.693$).

In terms of reaction times to accomplish embedded secondary tasks used to measure spare mental capacity, the results showed that at two points during the mission, operators in the Negative *A Priori* Priming group had significantly slower reaction times to a secondary task than operators in the Positive *A Priori* Priming group, as shown in Figure 54. Mann-Whitney comparisons showed that the Negative Priming group had 61% slower reaction times to a chat question in fifth minute compared to the Positive Priming control group ($Z = -3.115$, $p = 0.002$). Similarly, the Negative Priming group had 46% slower reaction times to a chat question in eleventh minute compared to the Positive Priming control group ($Z = -3.115$, $p = 0.002$). There were no significant differences in the other three secondary task measures among the *A Priori* Priming groups.

As shown in previous research (Cummings & Guerlain, 2004), an embedded secondary tool can provide an effective indicator of workload by measuring the spare mental capacity of the operator. These results could indicate that at certain points during the mission, operators in the Positive *A Priori* Priming group had more spare mental capacity than the Negative *A Priori*

166

Priming group. This higher level of spare mental capacity could potentially indicate that positive priming and higher trust in the AS caused operators to have lower workload.
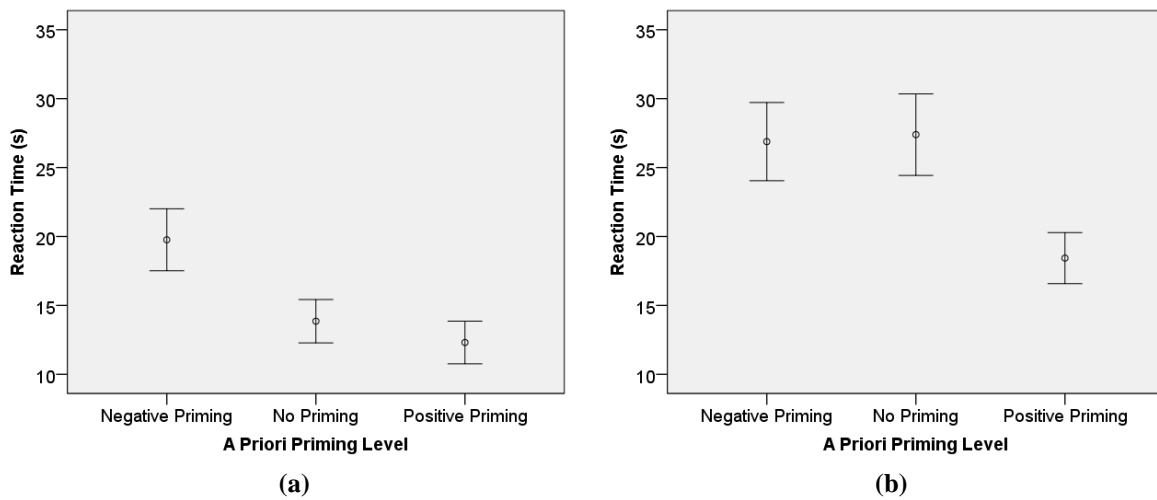
Figure 54. Reaction times to embedded secondary tasks comparison: (a) Chat question in fifth minute. (b) Chat question in eleventh minute. Standard Error bars are shown.

These results provide a few potential explanations for the mistakes in target destruction by the Negative *A Priori* Priming group. First, operators in the Negative *A Priori* Priming group had lower reported trust in the AS, which may have reflected their frustration with the system, leading to violation of the ROEs. A second possible explanation revolves around the fact that the Negative *A Priori* Priming group had lower reported expectations of system performance. These lower expectations could form a self-fulfilling prophesy, where operators expect to perform poorly and thus sabotage their own performance. However, all test subjects were instructed that following the ROEs was part of how they would be judged for the reward at the end of the experiment.

The third, more likely explanation is that if the Negative *A Priori* Priming group had less spare mental capacity, they may have misunderstood or missed the ROE instructing them not to destroy hostile targets until later in the mission. Similarly, they could have mistakenly designated an unknown target as hostile, leading to an erroneous hostile destruction. Many previous studies have demonstrated the negative impact of high cognitive workload on operator performance in human supervisory control of multiple UVs (Clare & Cummings, 2011; Cummings, Clare, et al., 2010; Cummings & Nehme, 2010; Dixon & Wickens, 2003; Ruff, et al., 2002) and this result could indicate that a similar situation was caused by negative priming.

5.7.2.2   Real-Time Priming

The goal of adjusting the Real-Time Priming level was to set operators' expectations of how well the system should perform. The intention was to induce the High Priming group into having higher expectations of performance as compared to their perception of how well the system was performing. According to the CHAS model, this higher Perceived Performance Gap (PPG) would lead to lower trust, a higher rate of intervention, and improved system performance.

Results showed that test subjects who experienced the Low Priming level had higher average perceptions of how well the system was performing. A Mann-Whitney comparison showed that the Low Priming group had 8.8% higher average real-time performance ratings compared to the High Priming group (Z = -2.122, p = 0.034), as shown in Figure 55a. This result was somewhat expected, as operators viewing the performance plot would likely think that the system was performing better when the reference line was lower. Surprisingly, the Low Priming group also had higher average real-time ratings of *expected* system performance. A Mann-Whitney comparison showed that the Low Priming group had 6.9% higher average expectations ratings compared to the High Priming group (Z = -2.145, p = 0.032), as shown in Figure 55b. There were no significant differences in trust ratings or calculated PPG between the Real-Time Priming groups.
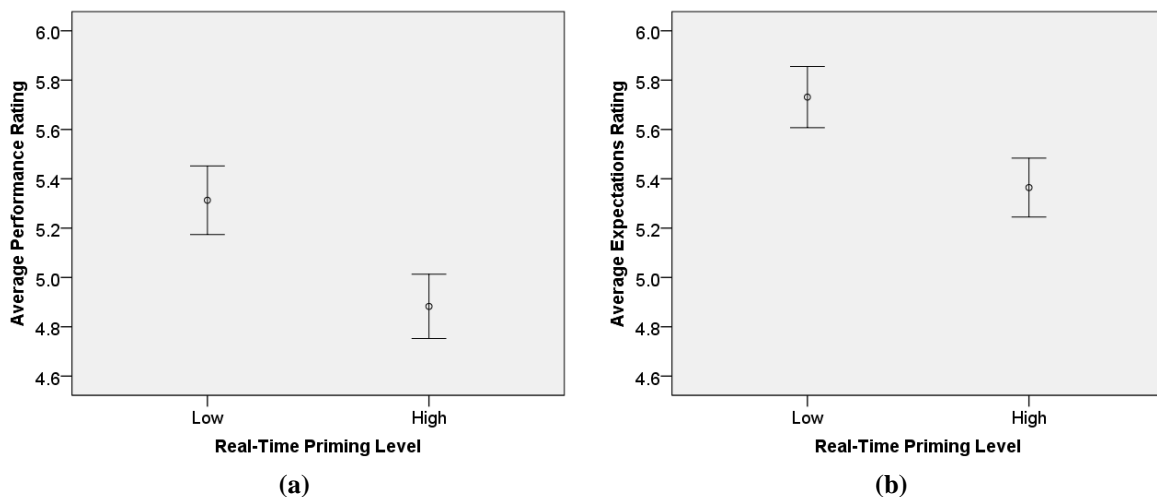
Figure 55. Average real-time ratings comparison: (a) Perceived system performance. (b) Expectations of system performance. Standard Error bars are shown.

In addition to comparing average ratings for the entire mission, changes in expectation and performance ratings over time were analyzed. A repeated measures ANOVA was utilized for this analysis, with a between-subjects factor of Real-Time Priming Level and a repeated measures factor of time, as shown in Figure 56. This ANOVA indicated a significant effect for Real-Time Priming Level, $F(1,82) = 4.590$, $p = 0.035$. There was also a significant effect for time ($F(8,656) = 3.386$, $p < 0.001$) and a significant interaction effect between time and Real-Time Priming Level ($F(8,656) = 2.118$, $p = 0.001$). It appears that both groups had similar perceptions of the performance of the system for the first half of the mission, but in the second half of the mission, the Low Priming group had significantly higher ratings of system performance than the High Priming group (Figure 56a). For ratings of expectations, the repeated measures ANOVA indicated a significant effect for Real-Time Priming Level, $F(1,82) = 4.140$, $p = 0.045$. This difference in expectations was consistent over time, with the Low Priming group reporting higher expectations of performance (Figure 56b).

(a)   **(b)**
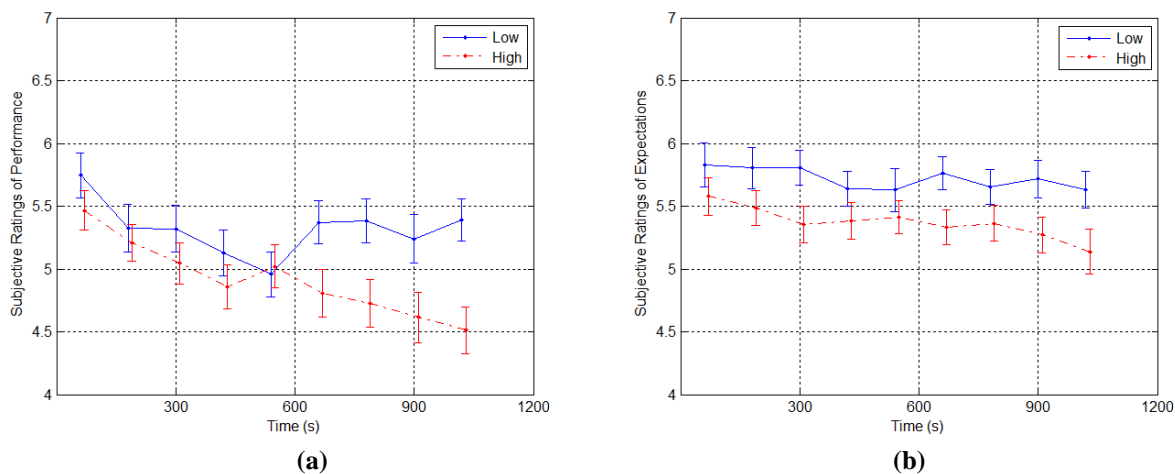
Figure 56. Real-time ratings comparison over time: (a) Perceived system performance. (b) Expectations of system performance. Standard Error bars are shown.

These results suggest that real-time priming had some of the desired effects on operator expectations and perceptions, but also led to unintended consequences. High Real-Time Priming did in fact lower operator perceptions of how well the system was performing. However, instead of raising operator expectations, higher real-time priming actually led to lower expectations of how well the system would perform throughout the rest of the mission. This likely reflects

operator frustration due to their perception of system performance compared to the reference line presented on the performance plot.

As will be discussed further in Section 5.7.4, operators in the High Priming level expressed frustration that they were struggling to keep up with the high reference area coverage curve shown on the plot. While there were no significant differences in total mouse clicks or other measures of operator interventions between the Real-Time Priming groups, some operators reported that they tried to intervene more frequently to improve system performance. However, others expressed that things seemed "out of their control," which may be reflected in the declining performance ratings towards the end of the mission (Figure 56a). Additional evidence of this frustration is that following each mission, the High Priming group had lower subjective ratings in response to the question, "How confident were you about your performance?" A Mann Whitney comparison showed that the High Priming group had 30% lower confidence ratings compared to the Low Priming group ($Z = -4.462$, $p < 0.001$). Operators likely lost confidence because they were unable to guide the suboptimal automated scheduling algorithm to achieve the high performance shown on the reference line in the performance plot.

While the original intention of higher real-time priming was to induce a higher Perceived Performance Gap (PPG) and lower trust in the AS, results showed that there were no significant differences in PPG or in trust ratings between the Real-Time Priming groups. This is likely due to the frustration and lack of confidence expressed by operators in the High Priming group. It is also possible that the real-time survey used to gather this subjective rating data was not sensitive enough to measure the differences in PPG or trust due to real-time priming.

In terms of system performance, there were no significant differences among the system performance metrics, including the primary metric of area coverage, across the Real-Time Priming groups. Operators in the High Real-Time Priming group were unable to do any better than their counterparts in the Low Real-Time Priming group, while being shown that their performance was below the reference line of how well previous operators had performed, driving frustration higher and confidence lower.

The only other significant difference among the dependent variables for the Real-Time Priming groups was in terms of reaction times to accomplish embedded secondary tasks. Of the five

embedded secondary tasks, the results showed that operators in the High Priming group had significantly faster reaction times to two secondary tasks as compared to operators in the Low Priming group. Mann-Whitney comparisons showed the that operators in the High Priming group answered a chat question in the seventeenth minute of the 20 minute mission significantly faster than operators in the Low Priming group ($Z = -3.019$, $p = 0.003$), as shown in Figure 57a. Also, following a chat message prompt to create a search task in a specific quadrant of the map during the sixteenth minute of the mission, the High Priming group created the search task significantly faster than the Low Priming group ($Z = -2.735$, $p = 0.006$), as shown in Figure 57b. There were no significant differences in the other three embedded secondary task reaction times, which occurred earlier in the mission. While these differences in reaction time only occurred during only two of the five embedded secondary tasks, it is still worth investigating why operators in the High Priming group had significantly faster reaction times.

(a)   (b)
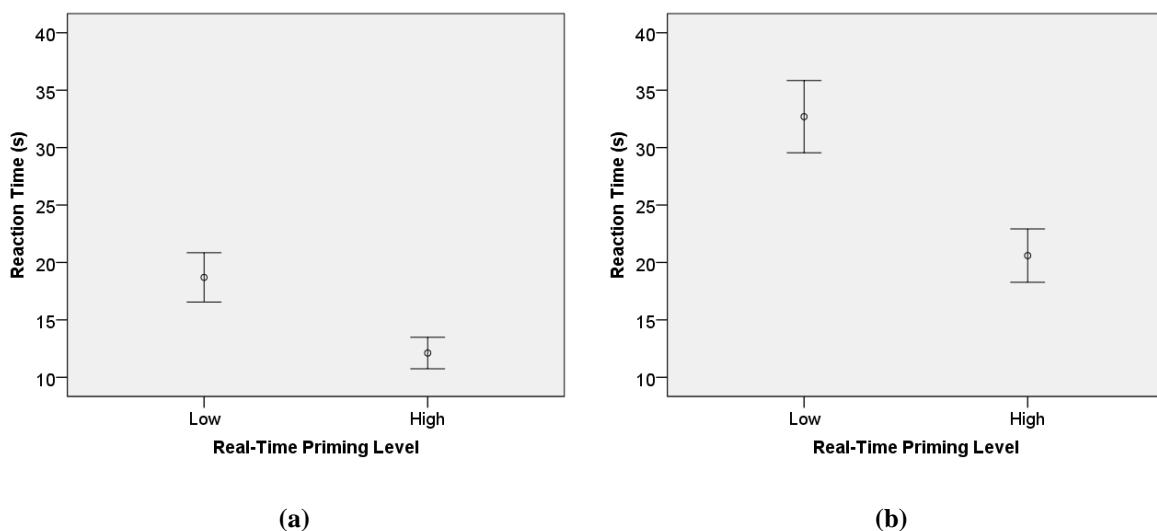
Figure 57. Reaction times to embedded secondary tasks comparison: (a) Chat question in seventeenth minute. (b) Prompted search task in sixteenth minute. Standard Error bars are shown.

There are three potential explanations for the faster reaction times of the High Priming group. First, it is possible, but unlikely that operators in the High Priming group had more spare mental capacity as compared to the Low Priming group. Both of these embedded secondary tasks occurred towards the end of the mission and frustration with their inability to keep up with the reference area coverage curve may have caused operators in the High Priming group to mentally "check out" from the experiment. However, this likely would not lead to faster reaction times to a chat message in the experiment. A second possibility is that in their attempts to improve system

171

performance, the High Priming group may have been more focused than operators in the Low Priming group, thus responding faster to chat message questions and prompts.

A third possible explanation is that the High Priming group may have chosen to focus their attention on the embedded chat tool due to their frustration with how the system was performing. The chat tool was visible from both the Map Display and the Schedule Comparison Tool (SCT), thus operators could continually monitor the chat tool. A previous study has demonstrated that operators may choose to fixate on a real-time chat secondary task instead of the primary supervisory control task (Cummings, 2004b). If the High Priming group was fixating on the chat tool in this experiment, then it would explain why the group had faster reaction times to chat message questions and prompts towards the end of the mission, when their ratings of system performance were at their lowest levels.

5.7.2.3   Information Time Delay

The goal of adjusting the Information Time Delay level was to evaluate the impact of a delay in the reporting of performance feedback on operator perceptions, operator behavior, and system performance. Results from the experiment showed that there were only two significant differences in the dependent variables between the Information Time Delay levels.

First, a repeated measures ANOVA showed that there was a significant three-way interaction effect for percentage of time that targets were tracked among all three independent variables (*A Priori* Priming Level, Real-time Priming Level, and Information Time Delay), $F_{(2,41)} = 3.992$, $p = 0.026$. It should be noted that there were no factor level or two-way interaction effects. This three-way interaction effect can be seen in Figure 58. Post-hoc Mann-Whitney dependent comparisons revealed two interesting results. The eight operators who had no *A Priori* Priming and the low reference line on the performance plot (Low Real-Time Priming) performed significantly worse in terms of percentage of time targets were tracked under the With Delay condition as compared to the No Delay condition ($Z = -2.521$, $p = 0.012$). In addition, the eight operators who had Positive *A Priori* Priming and the high reference line on the performance plot (High Real-Time Priming) also performed significantly worse in terms of percentage of time targets were tracked under the With Delay condition as compared to the No Delay condition ($Z = -2.380$, $p = 0.017$). A major caveat with these results is the small sample size, only 8 test subjects

in each of these specific conditions, conducting two missions each. However, it appears that adding a time delay to the performance plot negatively impacted system performance, but only under certain combinations of *a priori* and real-time priming. Notably, operators who experienced negative *a priori* priming had generally robust performance in terms of the percentage of time targets were tracked regardless of the information delay.

| (a) Low Real-Time Priming | (b) High Real-Time Priming |
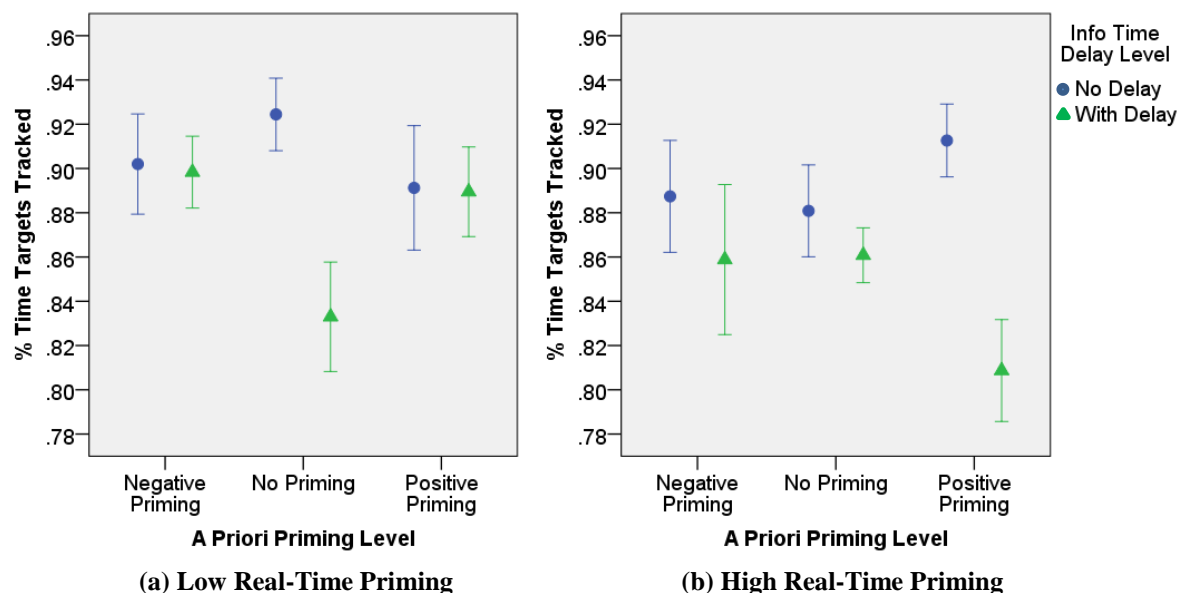
Figure 58. Three-way interaction effect for percentage of time that targets were tracked. (a) Low Real-Time Priming condition. (b) High Real-Time Priming condition. Standard Error bars shown.

The only other significant difference among the dependent variables between the Information Time Delay levels was in terms of Situation Awareness (SA). Chat messages provided intelligence information to the operator about whether "unknown" targets were actually hostile or friendly (based on their location on the map). It was up to the operator to re-designate these targets based on this information. Therefore, a second measure of SA was the ratio of correct re-designations of unknown targets to the number of unknown targets found. A Mann Whitney comparison showed that the No Delay missions had a higher average target re-designation accuracy (71.4%), while the With Delay missions had only 52.6% accuracy ($Z = -3.130$, $p = 0.002$).

Taken together, both results show that under the With Delay condition, operators did not perform as well in terms of tracking already found targets and re-designating unknown targets based on intelligence information, as compared to the No Delay condition. While adding an information

173

time delay did not lead to a statistically significant difference in the primary system performance metric of area coverage, results show that operators had lower system performance in terms of another measure and lower SA when there was an information time delay on the performance plot.

It is not immediately clear why the With Delay missions had lower system performance and lower SA than the No Delay missions. An analysis of the actions of operators (replan rate, search task rate, length of time spent replanning, utilization, etc.) in each Information Time Delay level did not reveal any statistically significant differences in operator behavior. There were also no statistically significant differences in the subjective ratings provided by operators. One of the limitations of this independent variable was that there was no way for the experimental testbed to gather precise data on human visual perception of information in order to measure the time delay in the perception of changes in system performance.

Finally, it should be noted that there were other ways for the operator to perceive area coverage performance beyond the performance plot. By default, operators had a "Fog of War" overlay on the Map View to indicate where they had recently searched (Section 3.2). Only 11 of the 48 test subjects ever turned the overlay off and in all cases it was turned back on and left on to the completion of the mission. Thus, the addition of an information time delay to the performance plot was not a pure information time delay; operators had alternative, albeit less accurate methods to estimate how well they were doing.

5.7.3   Demographic Predictors

A set of linear regression analyses was performed to see if there were any significant demographic predictor variables for high (or low) system performance, operator workload, and average trust. Details of the linear regression analysis can be found in Appendix T. There were four metrics with significant demographic predictor variables. In each case, there was only one significant predictor variable, which is equivalent to finding a significant linear correlation, and thus the results will be reported in this manner. It should be noted that there were no moderate to strong correlations among these four dependent variables, enabling separate analysis of each dependent variable.

For area coverage performance, the significant predictor variable was a metacognitive awareness score, ρ = -0.285, p = 0.005. A relatively weak negative relationship was found, indicating that operators with higher metacognitive awareness performed slightly worse in terms of the primary performance metric, area coverage. Metacognitive awareness was measured by test subjects' responses to a 52-question Metacognitive Awareness Inventory (MAI). Higher metacognitive awareness is typically associated with positive qualities, such as strategic planning for complex problem solving (Schraw & Dennison, 1994). However, it is possible that the rapid decision-making under uncertainty required for real-time human-automation collaborative scheduling relies more on developing heuristics through feedback on small adjustments rather than strategic and methodically-planned problem solving. Thus, it is debatable how useful MAI scores are for the type of decision-making necessary for this type of system.

For targets found, the significant predictor variable was self-rated frequency of watching television, ρ = -0.298, p = 0.003. Operators who reported that they watched more hours of television per day had lower system performance in terms of the number of targets found by the end of the mission. The literature is divided between whether television-viewing enhances creativity (Kant, 2012; Schmidt & Vandewater, 2008) or decreases cognitive development (Shejwal & Purayidathil, 2006). It is likely, however, that a passive activity such as watching television does not exercise the decision-making and attention allocation skills required to manage a team of UVs in a complex and dynamic environment, in contrast to video gaming, a more active form of recreation, which is discussed below.

For utilization, the significant predictor variable was self-rated frequency of playing computer and video games, ρ = -0.450, p < 0.001. This indicates a relatively strong relationship by human factors standards. As utilization measures the percent "busy" time of operators during the mission, this result shows that gamers were able to accomplish the mission with less "busy" time and thus more spare time for monitoring the system. It is likely that gamers in this experiment were able to click around the interface faster, accomplishing the same tasks in less time, which is reflected in the lower utilization level. In contrast to the previous finding on TV watching, video games are an active form of entertainment, and studies have shown that video game play improves visual attentional processing in divided attention, multi-tasking scenarios (Green & Bavelier, 2003). Another study showed that playing action video games improved encoding

speeds of visual information into visual short-term memory (Wilms, Petersen, & Vangkilde, 2013). Faster visual processing and encoding could lead to a decrease in utilization, as operators could comprehend the information presented to them faster in order to make quicker decisions. The impact of gaming frequency on operator trust and performance is explored further in Section 5.7.5.

Finally, for the 12-Question trust survey filled out at the end of the experiment, the significant predictor variable was whether or not the operator had ever or was currently serving in the military, $\rho = -0.308$, $p = 0.002$. The survey (Jian, et al., 2000) asked questions that measured both trust and distrust in the AS, with a total trust score ranging from -28 to 44. The 5 operators who had military experience had an average post-experiment trust score of 0.8, while the other 43 operators had an average trust score of 14.3. A Mann Whitney comparison showed that this difference was marginally significant ($Z = -1.723$, $p = 0.085$), and the trust ratings for both civilians and military test subjects are shown in Figure 59. It should be noted that of the 5 military test subjects, three experienced positive *a priori* priming and two experienced negative *a priori* priming.
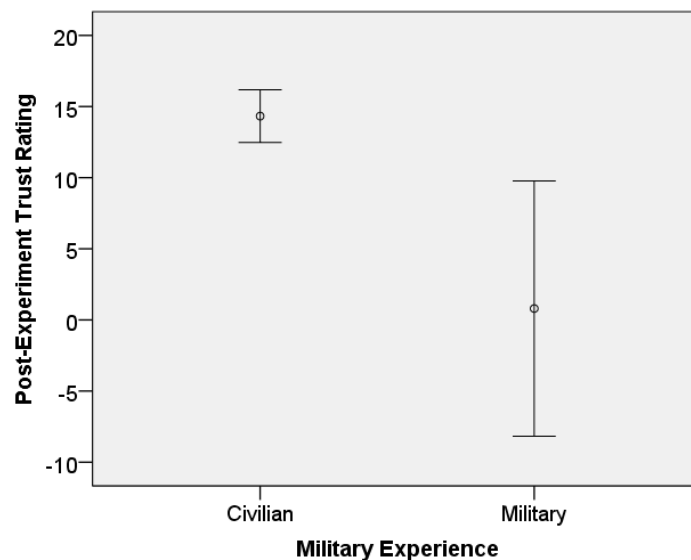
Figure 59. Post-experiment trust rating comparison by military experience. Standard Error bars shown.

While the sample size of military personnel is small, these results suggest that operators in the military may lose trust in automation faster than civilian operators when they perceive automation faults or poor automation recommendations. Previous studies have been inconclusive

176

on differences in military and civilian trust in automation. For example, in one study on maritime collision avoidance tools, preliminary interviews suggested that civilians were more receptive to proposed automated planning tools as compared to military personnel, but following testing of a prototype automation recommendation system, there were no statistically significant differences in trust ratings between the civilians and military personnel (Cummings et al., 2010). As Adams et. al (2003) discussed, both risk tolerance and organizational factors can influence trust. Military personnel understand that the decisions they are making could potentially impact the safety of their own team, jeopardize a critical mission objective, or waste valuable resources and expensive equipment. Under these circumstances, skepticism of an automated system is simply a method to mitigate risk. Adams et. al (2003, p. 67) further stated that "operators may be less likely to trust automation, a priori, if there is a prior track record of problematic automation being brought into [military] service." Historical military incidents involving automation, such as the Patriot missile system accidents (Cummings, Bruni, et al., 2010), can lower an operator's trust "inertia," leading to steep drops in trust following automation errors or poor automation performance.

Although significant correlations were found for these four demographic variables, causality is still unclear, and it will require further research to establish whether these demographic factors should influence training and selection processes for operators in real-time human-automation collaborative scheduling systems.

5.7.4 Qualitative Comments

Beyond quantitative subjective data, qualitative evaluations of the system and experiment were also obtained from all test subjects. While inferring operator perceptions and decision-making strategy from measureable actions during the missions and survey ratings is useful, valuable insights can also be gained from comments written by test subjects following the experiment.

Sixty-five percent of test subjects reported that the AS was fast enough for this dynamic, time-pressured mission. However, as in previous experiments (Clare, Cummings, How, et al., 2012; Clare, Hart, & Cummings, 2010), a common complaint from test subjects was a desire for increased vehicle-level control, as opposed to only goal-level control. Thirty-five percent of all test subjects wrote about wanting to manually assign vehicles to certain tasks because they

disagreed with an assignment made by the AS. For example, one operator wrote, "At times I wish I could have specifically assigned a vehicle to a search task directly," while another wrote, "Disliked that I could not assign tasks directly to a UV or force a UV to perform a certain task immediately." These comments could be due to the fact that the AS was taking into account variables that the human did not comprehend, such as the refueling needs, speed, or capabilities of each UV. Regardless of whether or not the AS was generating good schedules, operators *perceived* poor AS decisions and performance, supporting a key assumption in the CHAS model.

Another feedback survey question asked operators whether their trust in the AS increased or decreased over time. Forty-two percent of operators reported that their trust in the AS decreased throughout the mission, while 17% reported an increase in trust over time. The rest reported that they either had no change in trust (17%) or had both increases and decreases in trust (21%) throughout the mission, while 4% did not respond. There were no differences in these results based on the *A Priori* Priming level or Real-Time Priming level experienced by the operator. Examples of written comments include:

- "My trust in the scheduler increased when it successfully completed all the assigned tasks on time, but decreased when it did not."
- "[Trust] decreased, as high priority tasks were not always assigned by the automated scheduler. At times it favored lower priority tasks."
- "[Trust] decreased when it would fail to assign a task."
- "[Trust was] always pretty high."
- "There were times that my trust decreased because it did not assign the tasks I expected."
- "At certain points for area coverage it would schedule for a UV to move in directions that didn't seem logical. I would prefer it to search the west side if most of it had not yet been searched over the east."
- "My trust decreased every time I saw a path on the screen that I was not expecting. Usually I placed search tasks on the screen to influence the resulting path, but that did not always have the desired effect."
- "Trust increased with time as I got more comfortable and saw the Automatic Scheduler was working well."
- "Trust increased when it visibly showed how assignments changed according to plan priorities [changing the objective function of the AS]."
- "If there was a lot happening it was easier to just trust [the AS]."

These comments demonstrate that there was a wide a variety of opinions and perceptions of the AS across the test subjects. A number of insights into operator strategy and perceptions can be

drawn from these comments. First, the impact of human variability is clearly visible in these comments, as some operators gained trust over time while others lost trust, some thought the AS performed flawlessly while others could not understand the decisions made by the automation. Some operators reported that their trust would fluctuate, both rising and falling, depending on their perception of performance. This oscillatory behavior has been observed previously in the intervention actions of operators, as described in Section 4.2.1 and 4.2.3. An effective model of human-automation collaborative scheduling must have the ability to capture both the varying initial conditions of human operators, as well as the different possible ways that they could change their trust over time.

Second, the final comment shown above reflects the fact that under high workload situations, operators tend to rely on automation in order to reduce their workload. Whether or not this actually indicates an increase in trust is debatable, as it may simply be an increase in reliance due to cognitive overload. While the CHAS model captures the direct impact of high workload on the value that humans can add to system performance, this interesting interaction between high workload and reliance on automation is not currently captured in the model, but will be discussed in Chapter 7 under future work.

Third, these comments provide some interesting insights for the creators of scheduling algorithms and the designers of collaborative interfaces. As described in Chapter 1, there are often differences between the real world, the automation/engineer's model, and the human operator's models of the world. Generally, operators have expectations for how the automation should perform, for example which tasks should be assigned, what paths the UVs should take, and what search pattern should be used. When the automation chooses to do things in a different manner, even if this method is better in terms of some metric (distance traveled, priority level of tasks performed, fuel usage, etc.), it can confuse operators and decrease trust. Rather than training operators to understand the way the automation makes decisions, automation designers should develop methods that enable the automation to provide reasons for scheduling decisions, while interface designers should find innovate methods to display this information to operators. One method of providing insight into automation decision-making is direct-perception interaction (Gibson, 1979). As mentioned in one of the comments above, by allowing operators to immediately view the impact of changing the objective function of the AS (described further

in Section 3.2.1), it increased trust because the operator could infer some of the reasons behind the decisions made by the automation.

Finally, test subjects were asked about the performance plot and how it influenced their perceptions and behavior. They were not specifically asked about the time delay. Seventy-five percent of operators reported that they looked at the performance plot frequently, while 58% reported that they changed their behavior because of what they saw on the plot. There were no differences in these results between the Real-Time Priming levels, reinforcing the fact that real-time priming did not have the intended effect on operators. The intention was for the High Priming condition to cause operators to perceive a larger gap in performance and to more frequently intervene in the system as compared to operators in the Low Priming group. As described in Section 5.7.2.2, there were no significant differences in system performance or in any of the operator action measures between the Real-Time Priming groups.

However, the language used in comments made by operators in the different Real-Time Priming groups reveals the frustration created by the High Priming condition. Seventy-nine percent of operators in the High Priming group wrote about the pressure that they felt to improve their performance or the frustration that they felt over their poor performance in relation to the reference line, while only 13% of the operators in the Low Priming group expressed feelings of frustration or pressure. Forty-six percent of operators in the Low Priming group expressed that the performance plot made them feel good about their performance or that they felt they were doing well, while none of the operators in the High Priming group expressed these positive feelings. Examples of written comments about the performance plot include:

- High Real-Time Priming group
  - "It made me feel bad about myself. But more frustrating about it was that for the most part, the performance was based on the scheduler and a lot of it was out of my control (i.e. search) where I just had to sit and watch."
  - "It made me feel like I was underperforming by a large margin. I couldn't really alter my behavior (the output was different than expected), but the chart definitely added to my sense of frustration."
  - "It made me feel pressure to do better. I generally tried to increase my coverage area based on the plot."
  - "I gave up trying to use [the performance plot] since I was always below [the reference line]."

180

- o "I was always below [the reference line] so it made me want to constantly change my plans until it started increasing."
  - o "It made me realize I must be doing something wrong - which was area coverage (I was focusing on targets more)."
- Low Real-Time Priming group
  - o "It made me feel pretty good. Stayed above [reference line] - happy to see that!"
  - o "Whenever my actual performance was above the [reference line], I felt better about myself. When I saw my performance dropping, I tried to cover more area."
  - o "It made me feel good, I was ahead. I did not change my behavior, as I was always ahead of [the reference line]."
  - o "I was usually above average, so I guess I felt good about that."
  - o "I probably slacked off because of it though (as in I was calmer making decisions, but maybe could have been more vigilant)."
  - o "It made me feel good and in control to see the [performance] line was above [the reference line]. Once I saw that I was covering the area more than average, I focused more on tracking targets."

While the High Real-Time Priming condition may have caused operators to feel pressure to intervene more frequently in the system, this did not translate into measureable differences in operator actions or system performance. Instead, the frustration created by the High Priming condition was reflected both in these comments and in the lower ratings of confidence by operators in the High Real-Time Priming group (Section 5.7.2.2). It is likely that the reference line on the performance plot for the High Real-Time Priming condition was set too high, leading to a sense that the goal was unachievable. It also led to a shifting of responsibility, where some operators placed all blame for poor system performance on the automation. In contrast, the Low Priming condition had a low reference line on the performance plot which provided positive reinforcement to operators, increasing their confidence. Future work should explore whether a more achievable reference performance curve for the High Priming group could cause the intended effect of pushing operators to intervene more frequently without lowering operator confidence.

5.7.5 Gamer Analysis

A detailed analysis of the impact of frequency of computer and video game playing was conducted. This analysis was conducted for two main reasons. First, gaming frequency was the most significant demographic predictor of utilization, implying that frequent gamers could

181

operate the interface more quickly than nongamers and potentially had a lower workload level (Section 5.7.3). Second, although the experiment used random assignment of subjects to experimental conditions, the between-subjects condition of Real-Time Priming had an uneven balance of gamers and nongamers. A Mann-Whitney comparison showed that the average self-reported gaming frequency on a Likert scale from 1-5 (low to high) of the Low Priming group was significantly higher than subjects in the High Priming group (Z = -2.366, p = 0.018), as shown in Figure 60.
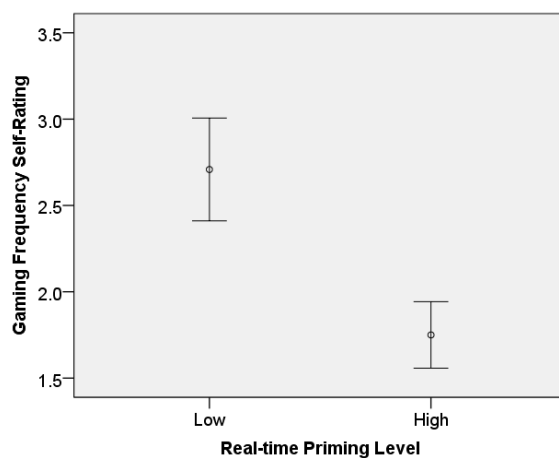
Figure 60. Uneven balance of gaming frequency between real-time priming groups.

These two findings encouraged a deeper analysis of the behavior of gamers. To begin, the analysis investigated whether gaming frequency correlated with performance metrics, behavior metrics, or subjective ratings from the experiment. Among the performance metrics, gaming frequency did not correlate with the primary performance metric of area coverage, $\rho = 0.008$, $p = 0.938$. Thus, for the primary performance metric in the CHAS model, there was no direct correlation with gaming frequency. However, gaming frequency did correlate with the percent of time that targets were tracked, $\rho=0.211$, $p=0.039$. This indicates that gamers were better at preventing targets from becoming lost by managing the resource allocation of the UVs.

As reported in Section 5.7.3, gaming frequency correlated with utilization, $\rho = -0.450$, $p < 0.001$. In addition, gaming frequency correlated with the average length of time to replan, $\rho = -0.286$, $p = 0.005$. Both of these results indicate that gamers were faster at using the interface, evaluating information, and making decisions. Also, in terms of operator behavior, there was a marginally significant correlation between gaming frequency and the total number of search tasks created, ρ

182

= -0.191, p = 0.062. Thus, it appears that gamers were less likely to intervene by creating new search tasks. In terms of subjective ratings, gaming frequency correlated with higher average real-time ratings of trust (ρ = 0.216, p = 0.037), perceived performance (ρ = 0.248, p = 0.016), and expectations (ρ = 0.302, p = 0.003). To summarize, these results indicate that gamers had higher trust in the AS and thus did not intervene as often. They were faster at using the interface and were better at preventing targets from becoming lost. They also had higher perceptions of how well they were doing and how well they expected to do.

While all of these correlations were weak to moderate in strength by human factors standards, the fact that gamers had a somewhat higher propensity to trust the automation raised the question of whether frequent gamers reacted to *a priori* priming differently than nongamers. To facilitate this analysis, test subjects were divided into categories of "gamers" and "nongamers" based on a test subject's self-reported frequency of playing computer and video games. Eighteen test subjects who reported that they were "weekly gamers," "a few times a week gamers," or "daily gamers" were classified as gamers. The other 30 test subjects reported that they played computer or video games once a month or less frequently. Test subjects were also asked to describe the types of games they played. Among the gamer group, all but one test subject reported playing an "action" video game, defined here as a shooter, real-time strategy, platform, racing, or sports game where the motion of a character or vehicle must be controlled directly and in real-time. While a previous study of the impact of video gaming used a more restrictive definition of action games, only counting shooter-type video games, the other types of games listed above can all "require fast reaction to multiple visual stimuli in a real time gaming environment" (Wilms, et al., 2013, p. 110). Future work should explore the differences in behavior and performance among gamers of different types, such as those who play shooter vs. real-time strategy games.

Beginning with the gamers, there was a fairly equal distribution of test subjects among the *A Priori* Priming levels: Negative (5), No Priming (7), and Positive (6). Each of these test subjects conducted two missions. Gamers who experienced the Positive Priming level had higher ratings of trust in the AS following their training mission, but prior to the actual experimental missions. Mann-Whitney pairwise comparisons showed that the Positive Priming gamers had 22.8% higher trust ratings compared to the No Priming gamers (Z = -2.598, p = 0.009) and 19.1% higher trust ratings compared to the Negative Priming gamers (Z = -2.364, p = 0.018). There were no

significant differences in pre-experiment trust ratings between the Negative Priming gamers and the No Priming gamers (Z = -0.278, p = 0.781). Similar results were found for the average real-time rating of trust during the missions, where the Positive Priming gamers had 23.9% higher trust ratings compared to the No Priming gamers (Z = -2.446, p = 0.014) and 35.2% higher trust ratings compared to the Negative Priming gamers (Z = -2.882, p = 0.002). Again, there were no significant differences in average real-time trust ratings between the Negative Priming gamers and the No Priming gamers (Z = -0.536, p = 0.592). After the end of the experiment, there were no significant differences in self-reported trust across the three *A Priori* Priming groups using the 12-question trust survey according to a Kruskal-Wallis omnibus test (χ^2(2, N=18) = 3.653, p = 0.161).

In terms of reported trust, these results are identical to the overall results presented in Section 5.7.2.1, in terms of the reaction to *A Priori* priming. Positive Priming had the desired effect of raising initial trust in the automation. This higher trust level among the Positive Priming gamers was maintained on average throughout the experiment. Negative Priming was not strong enough to significantly lower self-reported trust as compared to the control group. After the end of the experiment, there were no significant differences in self-reported trust, as the effects of priming did not endure.

Gamers began to differ from the overall test subject population in their system performance across the *A Priori* Priming groups. First, among the overall subject population, operators who experienced Negative *A Priori* Priming had a significantly higher number of mistakenly destroyed targets (Section 5.7.2.1). Secondary workload metrics showed that the Negative *A Priori* Priming group may have had less spare mental capacity during the mission. Thus, they may have misunderstood or missed the ROE instructing them not to destroy hostile targets until later in the mission or mistakenly designated a target as hostile. An analysis of the performance of gamers shows that five of the 14 missions with a target destruction mistake had a gamer as the operator. This is proportional to the number of gamers in the subject population (38%), thus gamers were no less likely to make a mistake as compared to nongamers. However, among gamers, a Kruskal-Wallis omnibus test showed that there were no significant differences in mistaken target destructions based on *A Priori* Priming level, χ^2(2, N=5) = 1.778, p = 0.411. One possible reason why gamers did not make more mistakes under the Negative *A Priori* priming

condition is that gamers had lower utilization levels than nongamers, as described above. Thus, they may have had more spare mental capacity to correctly interpret the ROEs and correctly designate targets.

A second difference in system performance between the general subject population and gamers is in the primary performance metric, area coverage performance. The overall test subject population had no significant differences in area coverage performance across the *A Priori* Priming groups (Section 5.7.2.1). However, a Mann-Whitney test showed that there was a marginally significant difference[1] between the Positive *A Priori* Priming gamers and Negative *A Priori* Priming gamers in terms of area coverage performance (Z = -1.715, p = 0.086). Gamers with Positive A *Priori* Priming had 35% higher average trust, but 10% lower average area coverage performance as compared to gamers with Negative A *Priori* Priming. These results are shown in Figure 61.
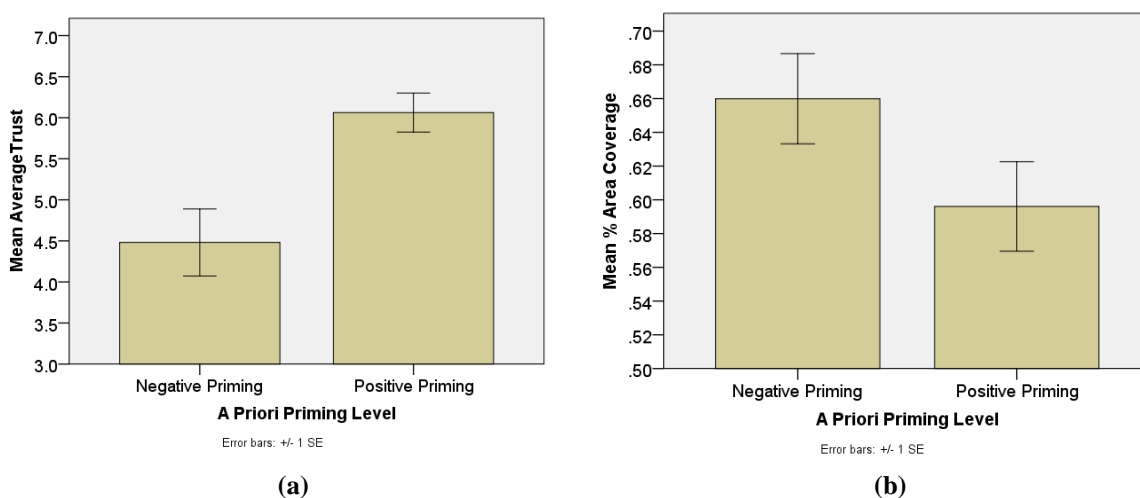
Figure 61. Impact of *a priori* priming on gamers. (a) Average real-time rating of trust in AS. (b) Area coverage system performance by the end of the mission.

An analysis of the actions of the gamers in these *A Priori* Priming groups provides some insight into why this difference in system performance occurred. Time series data on operator actions throughout the mission were compared using a repeated measures ANOVA. First, the Negative *A*

---

[1] A separate analysis of gamers who only experienced the Low Real-Time Priming condition showed that there was a significant difference between the Positive *A Priori* Priming gamers (8 missions) and Negative *A Priori* Priming gamers (8 missions) in terms of area coverage according to a Mann-Whitney test (Z = -1.997, p = 0.046).

185

Priori Priming gamers replanned more frequently than Positive *A Priori* Priming gamers ($F(1,20) = 7.147$, $p = 0.015$), as shown in Figure 8a. Second, in terms of the length of time that operators spent replanning, Negative *A Priori* Priming gamers spent marginally significantly less time evaluating new plans generated by the AS ($F(1,20) = 3.433$, $p = 0.082$), as shown in Figure 8b. Third, the Negative *A Priori* Priming gamers created search tasks more frequently throughout the mission ($F(1,20) = 5.045$, $p = 0.036$), as shown in Figure 8c.
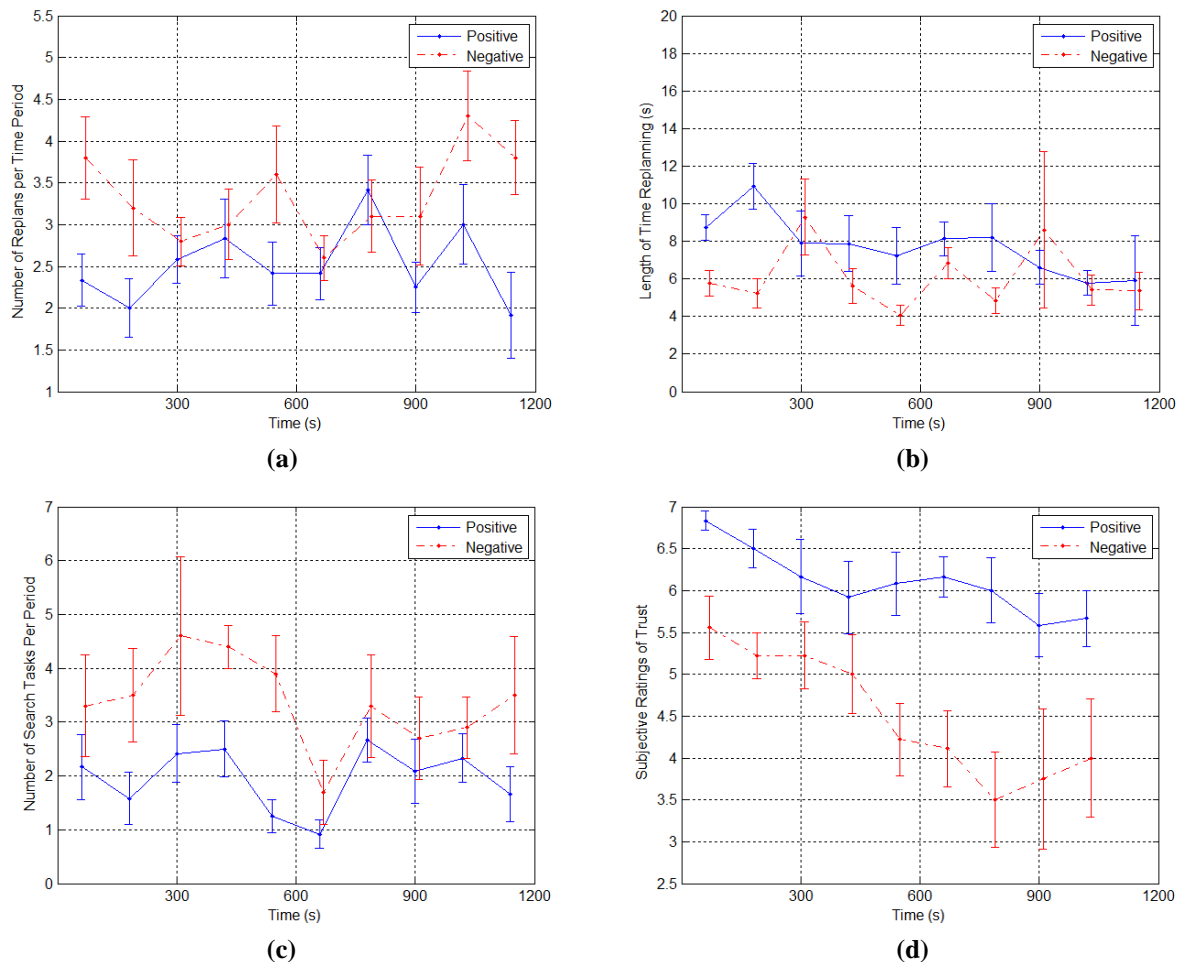
Figure 62. Differences in operator behavior between gamers who experienced positive or negative *a priori* priming. (a) Replan rate. (b) Length of time to replan. (c) Search task rate. (d) Trust ratings. Standard error bars shown.

Fourth, the Negative *A Priori* Priming gamers reported lower trust than Positive *A Priori* Priming gamers ($F(1,18) = 12.142$, $p = 0.003$), as shown in Figure 8d. There was also a significant effect for trust across time, indicating that trust decreased throughout the mission for both groups ($F(8,144) = 5.720$, $p < 0.001$). It should be noted that there were no significant

differences in utilization between the *A Priori* Priming groups for gamers, (F(1,20) = 1.016, p = 0.325), similar to the overall test subject results presented in Section 5.7.2.1.

Remarkably, gamers who experienced Negative A *Priori* Priming behaved in almost the exact same way as high performers in a previous experiment using this testbed (Section 3.2 and 3.3). Gamers in the Negative *A Priori* Priming group understood the imperfections in the automation, reporting lower trust in the AS. They modified their behavior appropriately and used the system as designed by replanning more frequently, spending less time evaluating new schedules generated by the AS, and creating more search tasks to encourage the UVs to explore new areas on the map. They were able to intervene at a higher rate without increasing their workload as compared to the gamers who experienced Positive *A Priori* Priming. In contrast, a similar analysis of nongamers revealed that while *A Priori* Priming did impact their reported trust in the AS, there were no significant differences in behavior or system performance across the *A Priori* Priming groups.

Why did *a priori* priming of trust only influence the behavior of *gamers* in a way that impacted system performance? One potential reason is that gamers may have a higher propensity to overtrust automation. As described above, gaming frequency correlated with higher average ratings of trust in the AS ($\rho = 0.216$, $p = 0.037$). Also, gamers began the mission with significantly higher ratings of trust as compared to all nongamers according to a Mann-Whitney test ($Z = -2.254$, $p=0.024$), but eventually adjusted their trust to the same, lower levels of trust of nongamers (Figure 63a).

Additional evidence of the different reactions of gamers and nongamers to *a priori* priming is provided by an analysis of total mouse clicks (descriptive statistics presented in Appendix T). This measure was not part of the original experiment design and its use is for post-hoc analysis only. An ANOVA indicated a significant difference in the total mouse clicks among the *A Priori* Priming Levels, $F(2,90) = 3.765$, $p = 0.027$. There was also a significant effect for gamer vs. nongamer: $F(1,90) = 4.458$, $p = 0.038$ and a significant interaction effect between gamer/nongamer and *A Priori* Priming Level: $F(2,90) = 4.135$, $p = 0.019$. Gamers overall had 11% fewer total mouse clicks on average as compared to nongamers. There was no difference in the total mouse clicks of nongamers across the *A Priori* Priming levels. However, gamers reacted

to Negative *A Priori* Priming with a significantly higher number of mouse clicks, while gamers with Positive A *Priori* Priming had the lowest number of mouse clicks of any group (Figure 63b). Gamers were prone to overtrusting the automation, and under Positive A *Priori* Priming, they did not take as much action to guide the automation.

(a)   **(b)**
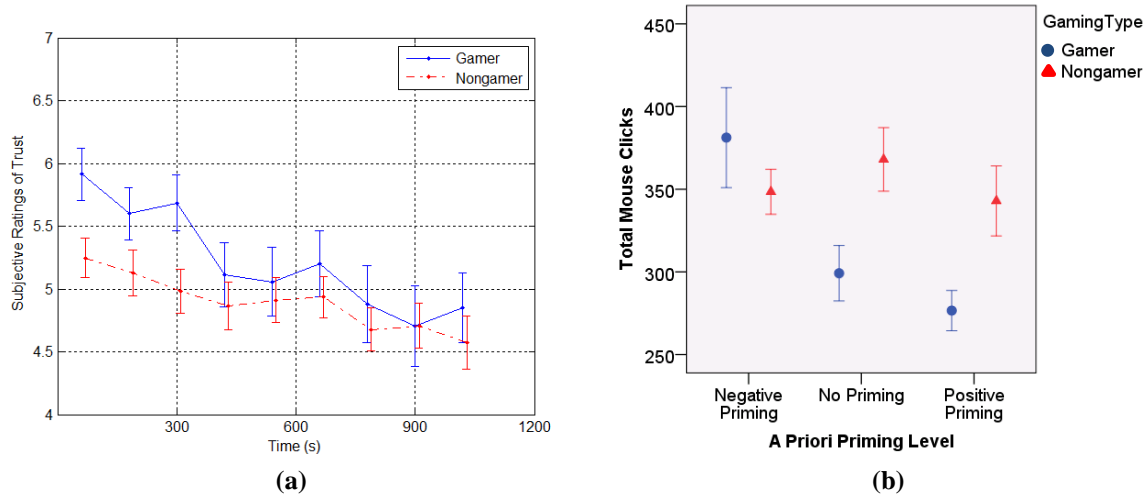
Figure 63. Comparison of gamers and nongamers: (a) Real-time trust ratings. (b) Total mouse clicks. Standard Error bars are shown.

To further analyze nongamers, a clustering analysis was conducted among all 30 nongamers, identifying 15 nongamers who reported high levels of initial trust in the AS in contrast to four nongamers who reported low initial trust. The clustering method was similar to the clustering analysis described in detail in Section 3.2.2, except that the clustering metric was initial reported trust in the AS. The average trust ratings in the AS for high and low initial trust nongamers are compared to the trust ratings of all gamers in Figure 64. High initial trust nongamers showed a decline in trust throughout the mission, similar to gamers. Low initial trust nongamers generally remained at a low level of trust throughout the mission. However, using the same repeated measures ANOVA method described above, it was found that there were no significant differences between high initial trust nongamers and low initial trust nongamers in terms of area coverage performance ($F(1,36) = 0.179$, $p = 0.675$), search task rate ($F(1,36) = 0.198$, $p = 0.659$), replan rate ($F(1,36) = 0.847$, $p = 0.364$), nor length of time to replan ($F(1,23) = 0.014$, $p = 0.906$). An ANOVA analysis similarly indicated that there were no significant differences in the total mouse clicks between high and low initial trust nongamers, $F(1,36) = 0.050$, $p = 0.825$. However, it should be noted that high initial trust nongamers had significantly lower utilization as compared to low initial trust nongamers according to a repeated measures ANOVA ($F(1,36) =$

4.969, p = 0.032), which indicates that high initial trust nongamers were less busy throughout the mission.
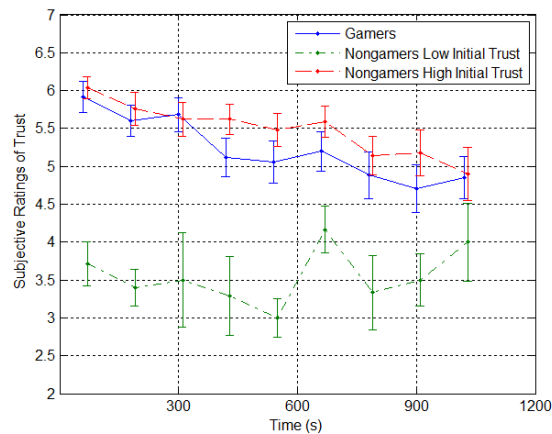
Figure 64. Real-time trust ratings for gamers, high initial trust nongamers, and low initial trust nongamers. Standard Error bars are shown.

Overtrust in the AS can lead to the phenomenon of automation bias (Mosier, et al., 1998), where operators disregard or do not search for contradictory information in light of an AS-generated solution which is accepted as correct (Cummings, 2004a). A number of empirical studies have shown that when working with imperfect automation, automation bias occurs (Chen & Terrence, 2009; Lee & Moray, 1994; Muir & Moray, 1996; See, 2002). Thus, Positive A *Priori* Priming may have induced automation bias in gamers, where they overtrusted the imperfect AS in this experimental testbed and did not intervene frequently enough. Negative *A Priori* Priming may have pushed gamers to a more appropriate level of trust in the AS, helping them avoid automation bias and encouraging them to intervene more frequently.

Another possible reason is that priming triggered a learned behavior in gamers. Wickens and Hollands (2000) proposed in their human information processing model that working memory has a more direct impact on perception and response selection as compared to long-term memory. Previous studies have shown that priming can "spread activation," meaning that the prime activates an association in memory prior to carrying out a task (Anderson, 1983; Niedeggen & Rösler, 1999). Additionally, the strength of the effect of the prime on behavior is influenced by the match between earlier experiences and the current situation (Domke, Shah, & Wackman, 1998; Lorch, 1982). Gamers, especially those who play action video games, have learned how to manually control characters or vehicles and manipulate automation to obtain a

desired result. Thus, Negative A *Priori* Priming of gamers may have activated this previously learned behavior to intervene more frequently in order to manipulate the automation to improve performance. In contrast, nongamers likely did not have as much previous experience working with automation and thus their behavior did not change significantly even though they reported lower trust.

These results have interesting implications for personnel selection and training for future real-time human-automation scheduling systems for multiple UVs. While gamers may bring valuable skills, such as faster visual attentional processing (Green & Bavelier, 2003) and faster encoding of visual information into short-term memory (Wilms, et al., 2013), they are also potentially prone to automation bias. One potential method for overcoming this propensity to overtrust automation is through priming during training. Results in this experiment demonstrated that the effects of priming are not enduring, thus regular priming throughout missions may be necessary to maintain the appropriate level of trust, as Rice et. al (2008) proposed.

While these results are compelling, the limitations of this analysis must be taken into account. The definition of a "gamer" is based on self-reported information, meriting further research to establish which types of video games and what frequency of video game play influence operator trust, behavior, and performance. The sample size of gamers in this data set was small, only 18 test subjects conducting two missions each. Future research should aim to evaluate these findings with a larger sample size of gamers and nongamers. Despite these limitations, initial evidence shows that previous experiences with automation and video game play can have a significant impact on initial trust level in automation and reaction to priming/training methods.

5.7.6 Time Series Data

One of the major objectives of this experiment was to gather data to evaluate the assumptions in the CHAS model surrounding perceptions of performance, expectations of performance, trust, and workload. Aggregate time series data from all test subjects was evaluated using a repeated measures ANOVA to test these assumptions. The results are presented below.

Operators were asked to rate their perception of performance, expectations of how well the system should be performing, and trust in the AS every two minutes throughout the mission. The

survey utilized a Likert scale of 1-7 (low to high). The aggregate ratings for all test subjects are shown in Figure 65. Several observations can be made from this data. First, operator expectations were significantly higher than perceived system performance throughout the mission, $F_{(1,166)} = 9.279$, $p = 0.003$. This supports the assumption made in the CHAS model that operators typically have higher expectations than perceptions of performance, leading to a positive Perceived Performance Gap (PPG).
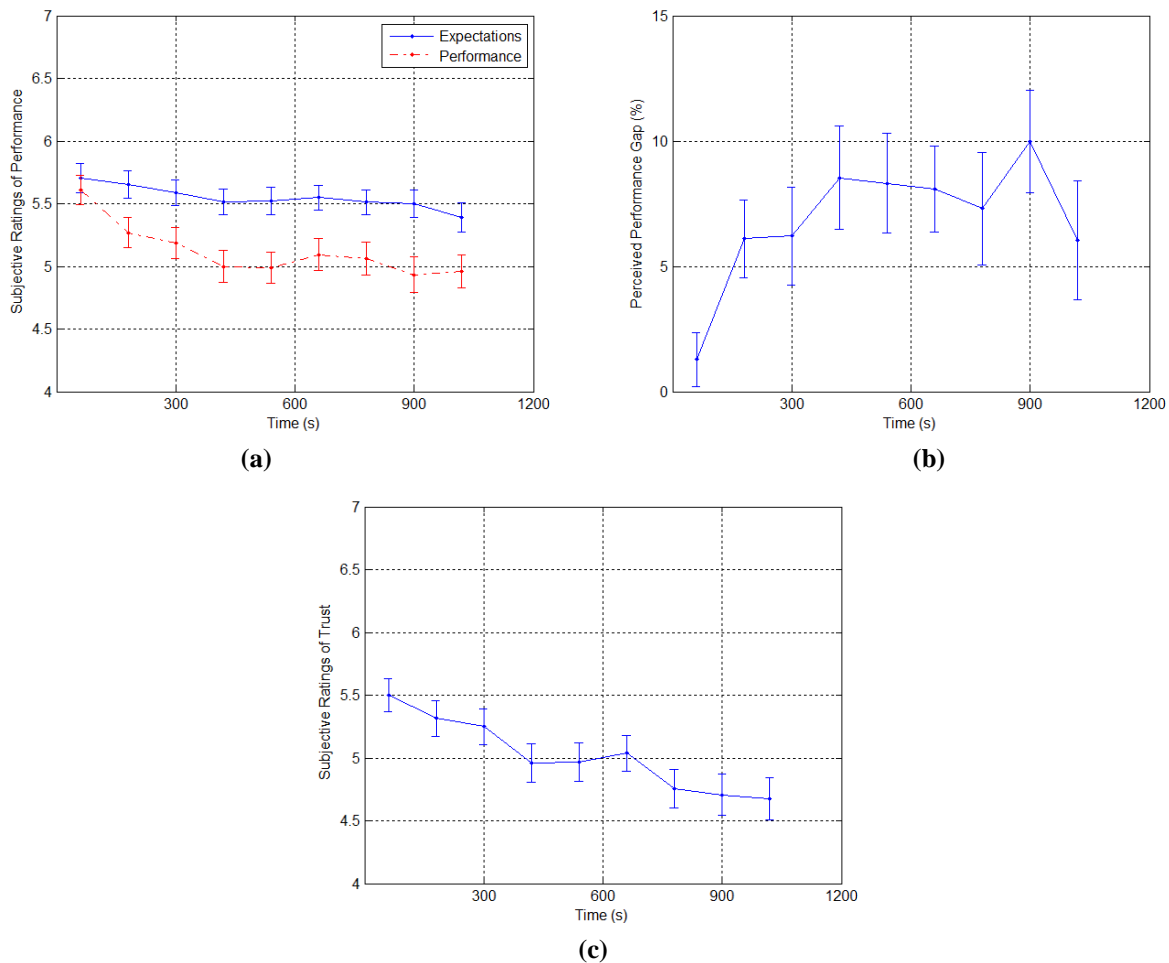
Figure 65. Aggregate real-time ratings (1-7, low to high) throughout the mission. (a) Expectations of performance and perceptions of performance. (b) Perceived Performance Gap (PPG), equal to the percent difference between ratings of expectations of performance and perceptions of performance. (c) Trust in the AS. Standard error bars are shown.

Second, operator perception of performance declined throughout the mission, as shown by the red line in Figure 65a. The repeated measures ANOVA indicated a significant time effect for perception of performance, $F_{(8,664)} = 5.083$, $p < 0.001$. This supports the CHAS model

191

assumption that the operator correctly perceives declining system and automation performance over time as the mission proceeds. It should be noted that the CHAS model can be tuned to a system which maintains or even improves in performance over time. Third, operators lowered their expectations based on their perception of system performance. A repeated measures ANOVA indicated a marginally significant time effect for expectations of performance, $F(8,664) = 1.811$, $p = 0.072$. The fact that operators adjusted their expectations of performance supports the need for an Expectations Adjustment feedback loop in the CHAS model, rather than assuming that the operator's expectations of performance are static.

Fourth, operators adjusted their expectations slowly in comparison to their perception of performance, as shown in Figure 65a. This is supported by a fourth metric, Perceived Performance Gap (PPG), which was calculated by taking the percent difference between the expectation and performance ratings at each two minute interval. There was a significant time effect for PPG, $F(8,664) = 2.519$, $p = 0.011$, indicating that the PPG increased over time, eventually reaching a 10% difference between expectations and performance ratings (Figure 65b). This supports the need to separately model the time delays for perceiving performance versus adjusting expectations.

Fifth, operator trust in the AS generally declined throughout the mission, as shown in Figure 65c. There was a significant time effect for trust, $F(8,664) = 8.528$, $p < 0.001$. The CHAS model assumes that human trust in the AS is negatively dependent on the PPG. This data analysis demonstrates that PPG generally increased over time as trust was declining. While this analysis alone cannot conclusively support the causal relationship between PPG and Human Trust, the qualitative comments described in section 5.7.4 provide additional evidence that as operators perceived that the AS was performing below their expectations, they lost trust in the AS.

Finally, one of the main objectives of this experiment was to evaluate the dynamic hypothesis of the CHAS model: if operators can either a) anchor to the appropriate trust in the AS and expectations of performance earlier in the mission and/or b) adjust their trust and expectations faster through better feedback about the AS, then system performance should improve. To evaluate this hypothesis, a cluster analysis was conducted to identify the missions which had significantly high or low performance, using total area coverage by the end of the mission as the

clustering metric. Details of the clustering method can be found in Section 3.2.2, where the same clustering method was used for data analysis with a previous data set. Of the total 96 missions, there were 17 missions in the High Performance cluster and 15 missions in the Low Performance cluster. High performers averaged 75.6% area covered by the end of the mission, while low performers averaged 49.6%. The trust ratings in the AS for High and Low performing missions are compared in Figure 66.

A repeated measures ANOVA indicated that there was a significant interaction effect for trust between time and performance cluster, $F_{(8,200)} = 2.768$, $p = 0.006$. The main effects for performance cluster ($F_{(1,25)} = 1.103$, $p = 0.304$) and time ($F_{(8,200)} = 1.875$, $p = 0.066$) were not significant. It appears that high performers anchored to a lower level of trust in the AS and essentially remained at the same level of trust for the entire mission. Low performers, in contrast, began at a higher level of trust and adjusted their trust over time, reducing it to the point where there were no statistically significant differences in trust as compared to the high performers.
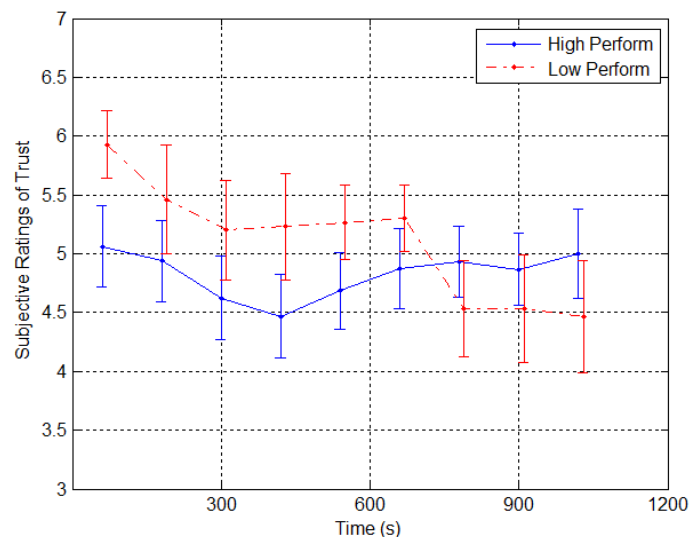
Figure 66. Comparison of real-time ratings of trust in the AS (1-7, low to high) throughout the mission between high and low performers. Standard error bars are shown.

This data provides evidence which supports the overall dynamic hypothesis of the CHAS model. High performers anchored to the appropriate level of trust early in the mission, understanding the imperfections in the automation and compensating to improve system performance. Low performers anchored at a higher level of trust, and while they did adjust their trust over time, by

the time they fully understood the limitations of the automation, it was too late in the mission to improve their performance.

5.8 Evaluation of Hypotheses and Model Predictions

Given all of the results presented in Section 5.7, the experimental hypotheses described in Section 5.2 can be evaluated. Specifically, the CHAS model made quantitative predictions of the impact of changes in system design and operator training on human and system performance. The experiment results are compared to the predictions made by the CHAS model below.

5.8.1 *A Priori* Priming

For the *A Priori* Priming independent variable, Hypothesis 1 stated that negative *a priori* priming of human trust in the AS was expected to result in a 9% increase in system performance by the end of the mission. Also, Hypothesis 2 stated that positive *a priori* priming of human trust in the AS was expected to result in a 4% decrease in system performance by the end of the mission. Finally, Hypothesis 3 stated that negative *a priori* priming of human trust in the AS was expected to result in a 12% increase in over the positive priming condition.

While *a priori* priming was successful at adjusting initial human trust in the AS, data from the general test subject population did not support these hypotheses. According to a repeated measures ANOVA, there were no statistically significant differences in the primary performance metric, area coverage, across the *A Priori* Priming levels, $F(2,41) = 0.016$, $p = 0.984$ (see Table 28 in Appendix T for descriptive statistics). The only statistically significant difference in system performance was that operators in the Negative *A Priori* Priming group had lower performance in terms of mistakes in target destruction. It is possible that the Negative *A Priori* Priming group had less spare mental capacity and thus misunderstood or missed the ROE instructing them not to destroy hostile targets until later in the mission (Section 5.7.2.1). This contradicts Hypotheses 1 and 3, as negative *a priori* priming may have caused a significant decrease in one of the system performance metrics.

When controlling for gaming frequency, however, the experimental data provides support for Hypothesis 3 and the CHAS model quantitative predictions. As described in Section 5.7.5, among gamers, there were no statistically significant differences in mistaken targets destroyed

194

across the *A Priori* Priming groups. Gamers had more spare mental capacity because they were faster at using the interface, evaluating information, and making decisions. Most importantly, gamers who experienced negative *a priori* priming had 10% higher average area coverage performance (M = 66.0%, SD = 8.4%) as compared to gamers with positive *a priori* priming (M = 59.6%, SD = 9.2%), a marginally statistically significant difference, which directly supports Hypothesis 3 (Figure 67). While the sample size of gamers was small and differences in system performance from the control No Priming group were not statistically significant, there is some evidence that negative priming can improve system performance while positive priming can reduce system performance.
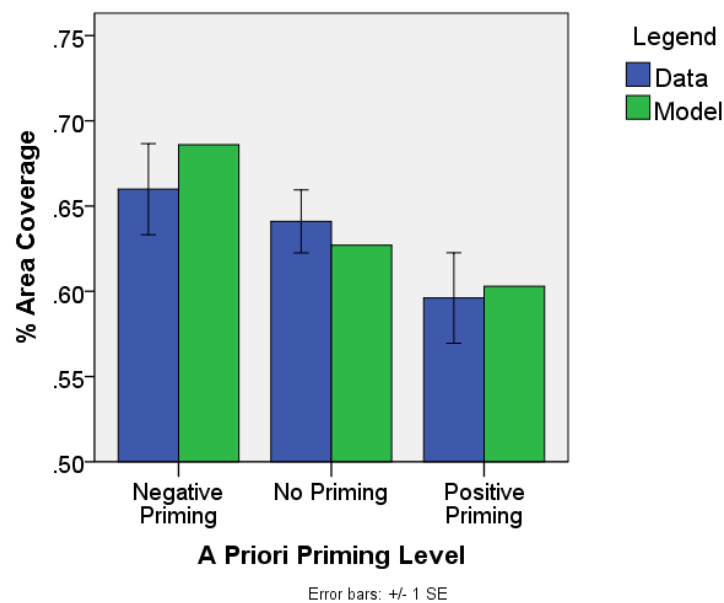
Figure 67. Predictions using the CHAS model compared to experimental results for gamers.

The quantitative model predictions presented in Section 5.2.1 are compared to experimental results from gamers in Figure 67. As shown, the model effectively captures the impact of changes in initial trust on area coverage performance for gamers. All predictions fall within ±1 Standard Error of the experimental data. These results provide compelling evidence that the model can accurately predict the impact of a change in operator training on system performance. A major caveat is that the model's predictions were only accurate for the gamer population of test subjects. Controlling for gaming frequency reduced some of the human variability in the experimental results. Future research should explore whether the model can make accurate

predictions for a more general population of operators given a different change in operator training that has a stronger effect on operator perceptions and behavior.

5.8.2 Real-Time Priming

For the Real-Time Priming independent variable, Hypothesis 4 stated that high real-time priming of operator expectations of performance was expected to result in an 8% increase in system performance by the end of the mission over low real-time priming. Results from the experiment did not support this hypothesis, as there were no significant differences among the system performance metrics, including the primary metric of area coverage, across the Real-Time Priming groups (see Table 29 in Appendix T for descriptive statistics).

Results showed that real-time priming had some of the desired effects on operator expectations and perceptions, but also led to unintended consequences (Section 5.7.2.2). Instead of raising operator expectations, higher real-time priming actually led to lower expectations of how well the system would perform throughout the rest of the mission. In the second half of the mission, the Low Priming group had significantly higher ratings of system performance than the High Priming group. Both of the results likely reflect operator frustration due to their perception of system performance compared to the reference line presented on the performance plot. Additionally, the High Priming group had 30% lower confidence ratings compared to the Low Priming group following the mission. The written comments of operators in the High Priming group express frustration and a lack of confidence (Section 5.7.4). It is possible that due to this lack of confidence, the frustration experienced by operators in the High Priming group did not translate into significant differences in Perceived Performance Gap (PPG), trust, or system performance. It is also possible that the surveys and measures of system performance failed to detect a change in these quantities due to the different Real-Time Priming levels.

The CHAS model assumes for the High Priming prediction that operator expectations of performance are 56% higher than in the Low Priming condition (Section 5.2.2), which should cause an increase in the operator's Perceived Performance Gap (PPG). The High Priming group had an average PPG that was only 13% higher than the PPG of the Low Priming group (see Table 38 in Appendix T for descriptive statistics), although this difference was not significant according to a Mann-Whitney test ($Z = -0.314$, $p = 0.753$). The model overstated the impact of

Real-Time Priming on system performance because Real-Time Priming did not have the desired effect on operator expectations. To further evaluate the CHAS model's accuracy, the model was run with only a 13% increase in expectations from the Low to High Priming conditions. The model results are compared to experimental results in Figure 68. As shown, the model effectively captures the impact of the actual changes in expected performance on area coverage performance. All predictions fall within ±1 Standard Error of the experimental data.
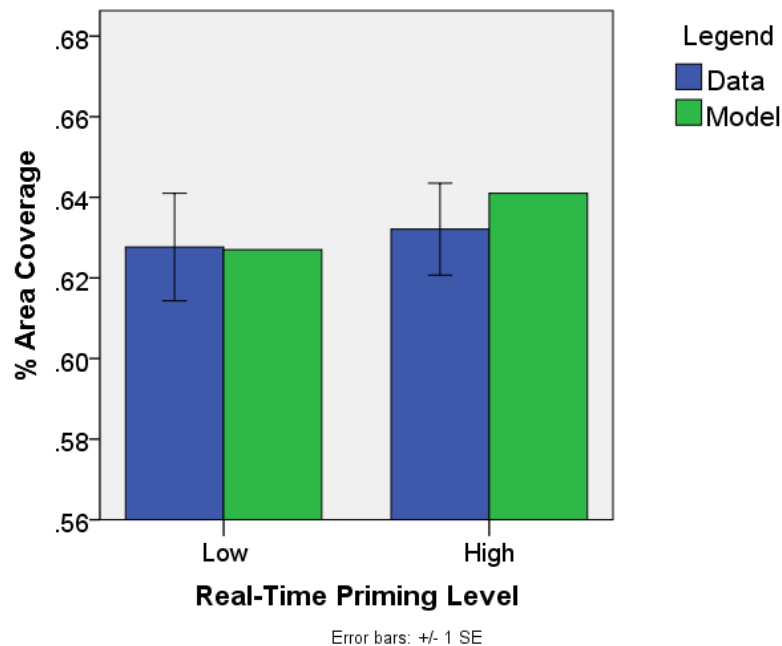
Figure 68. Revised predictions using the CHAS model compared to experimental results for all test subjects.

However, as stated earlier, according to a repeated measures ANOVA, there were no statistically significant differences in area coverage between the Real-Time Priming levels, $F(1,41) = 0.002$, $p = 0.961$. Thus, future research should explore whether a more achievable reference line on the performance plot for the High Priming condition could cause the intended effect of pushing operators to intervene more frequently without lowering operator confidence. Also, a different method of priming expectations of performance could be explored to further investigate the impact of changes in expectations on system performance.

5.8.3 Information Time Delay

For the Information Time Delay independent variable, Hypothesis 5 stated that the addition of an information reporting time delay to the performance plot was expected to result in a 5% decrease

in system performance by the end of the mission. Results from the experiment both support and contradict this hypothesis. Under certain combinations of the two priming independent variables (Section 5.7.2.3), the With Delay missions had significantly lower system performance in terms of the percentage of time that targets were tracked (see Table 30 in Appendix T for descriptive statistics). Also, operators in the With Delay missions had significantly lower Situation Awareness (SA) as measured by the ratio of correct re-designations of unknown targets to number of unknown targets found. Both of these results support Hypothesis 4. However, in terms of the primary performance metric of area coverage, there were no statistically significant differences in area coverage between the Information Time Delay levels according to a repeated measures ANOVA, $F(1,41) = 0.495$, $p = 0.486$.

In retrospect, the implementation of a time delay in this manner was fraught with challenges. First, as noted in Section 5.2.3, the predicted difference in system performance was small (5%) and thus it was likely that the experimental data would not have statistically significant differences. Also, the CHAS model assumed for the With Delay prediction that operators would perceive changes in the area coverage rate with a time delay of 130 seconds compared to only a 10 second delay for the No Delay condition (Section 5.2.3). However, there was no way for the experimental testbed to gather precise data on human visual perception of information in order to measure the actual time delay in the perception of changes in system performance. Future research could examine this with eye tracking devices for a more precise estimate.

Another major limitation to this implementation of a time delay was that there were other ways for the operator to perceive area coverage performance beyond the performance plot. Operators could view a "Fog of War" overlay on the Map View to indicate where they had recently searched (Section 3.2). Thus, the addition of an information time delay to the performance plot was not a pure information time delay; operators had alternative, albeit less accurate methods to estimate how well they were doing. Future research could run a similar experiment without the "Fog of War" overlay.

Despite these challenges, modeling time delays in operator perception of performance is crucial to the validity of the CHAS model. In an attempt at model simplification, the time delay in perception of system performance was removed from the model. However, the model was no

longer able to replicate the goal-seeking and oscillatory behavior that was observed in previous experimental data (Sections 4.2.1 and 4.2.3). Sterman (2000, p. 663) wrote that "oscillation requires…that there be time delays in the negative feedbacks regulating the state of a system." Previous research has shown that information time delays can have serious consequences for human decision-making in dynamic systems (Brehmer, 1990; Sterman, 1989b). Thus, it is crucial to capture delays in human perception of system performance when modeling human-automation collaborative scheduling of multiple UVs. Future research should explore whether a different system design change representative of real-world communications delays, such as delaying all information displayed to the operator (including visual information on the Map Display) similar to (Walker, et al., 2012), has a significant impact on operator behavior and performance in the OPS-USERS test bed.

5.9 Summary

This chapter described a human subject experiment that was conducted to evaluate the ability of the Collaborative Human-Automation Scheduling (CHAS) model to predict the impact of changes in system design and operator training on human and system performance. Results from the experiment led to a number of interesting findings. Eighty-three percent of operators were able to improve performance as compared to the performance of the system without human guidance, with an average improvement in performance of 12%. This aligns with previous results using this testbed and adds additional support to the assumption that the human operator can add value over the automation generated performance. Next, priming the initial trust level of operators using quotes from previous users of the system was successful at adjusting initial trust levels, with the strongest effect from positive priming. Test subjects who play computer and video games frequently were found to have a higher propensity to over-trust automation, but also experienced lower workload levels. By priming these gamers to lower their initial trust to a more appropriate level, system performance was improved by 10% as compared to gamers who were primed to have higher trust in the AS. The CHAS model accurately predicted the impact of this change in operator training on system performance.

Two other system design changes to provide feedback on performance were implemented in the experimental testbed. These two changes were implemented in the performance plot of the main

display and consisted of a) displaying different reference system performance "expectation" lines and b) implementing a time delay in the reporting of the actual system performance thus far in the mission. However, results showed that they did not have the intended impact on operator expectations and perceptions. Revised model predictions taking into account the actual changes in operator expectations and perceptions due to these system design changes resulted in more accurate predictions of system performance.

Both quantitative and qualitative data from this experiment validated a number of the assumptions in the CHAS model and provided some interesting insights for the creators of scheduling algorithms and the designers of collaborative interfaces. Operators adjusted their expectations of performance over time, supporting the need for an expectations adjustment feedback loop in the CHAS model. Results showed that as operators perceived that the AS was performing below their expectations, they lost trust in the AS. Operator comments suggested that rather than training operators to understand the way the automation makes decisions, automation designers should develop methods that enable the automation to provide reasons for scheduling decisions, while interface designers should find innovate methods to display this information to operators. Finally, experimental data supported the overall dynamic hypothesis of the CHAS model, showing that high performers anchored to the appropriate level of trust early in the mission, understanding the imperfections in the automation and compensating to improve system performance.

Having built confidence in the CHAS model's assumptions and accuracy, Chapter 6 will explore potential uses for the CHAS model by system designers. The CHAS model will be compared with another simulation model of human supervisory control of multiple UVs. Finally, the model's generalizability to other real-time human-automation scheduling systems will be discussed and the limitations of the model will be described.

6 Model Synthesis

Designers of UV systems currently have few tools to address some of the common challenges in human-automation collaboration that were identified in Chapter 1, which included inappropriate levels of operator trust, high operator workload, and a lack of goal alignment between the operator and AS. To attempt to address these issues, researchers and designers test different system components, training methods, and interaction modalities through costly human-in-the-loop testing.

The purpose of the Collaborative Human-Automation Scheduling (CHAS) model is to aid designers of future UV systems by simulating the impact of changes in system design and operator training on human and system performance. The goal is to reduce the need for time-consuming human-in-the-loop testing that is typically required to evaluate such changes. Also, designers can utilize this model to explore a wider trade space of system changes than is possible through prototyping or experimentation.

This chapter presents four example use cases of the CHAS model, to illustrate how it could aid UV system designers. Then, the CHAS model's accuracy and features are compared with a previously developed Discrete Event Simulation (DES) model of human supervisory control of multiple UVs. Finally, the generalizability of the model is discussed along with model limitations.

6.1 Potential Uses for CHAS Model

Four example applications of the CHAS model are presented to illustrate how the model could potentially be used by UV system designers. First, designers can use the model to further investigate the impact of operator trust in the Automated Scheduler (AS) on system performance. Second, the CHAS model can be used to explore a wider system design space that includes both traditional system components as well as human characteristics. Third, the CHAS model can support requirements generation for meeting system design specifications, such as maximum workload levels. Finally, designers can evaluate the impact of automation characteristics, such as the need for certain algorithms to have time to reach consensus, on human behavior and system performance.

6.1.1 Investigating the Impact of Trust on Performance

The experiment presented in Chapter 5 demonstrated that initial operator trust in the AS can have a significant impact on system performance. Specifically, test subjects who play computer and video games frequently were found to have a higher propensity to over-trust automation. By priming these gamers to lower their initial trust to a more appropriate level, they intervened more frequently to guide the suboptimal AS, improving system performance by 10% as compared to gamers who were primed to have higher trust in the AS.

This raises the question: what is the optimal human trust level and intervention rate when collaborating with a suboptimal algorithm? As described more extensively in Chapter 2, both overtrust and undertrust in automation can be detrimental to system performance. Low human trust in the AS can be caused by automation brittleness and operators with low trust may spend an excessive amount of time replanning or adjusting the schedule. On the other hand, overtrust in automation has been cited in a number of costly and deadly accidents in a variety of domains. Overtrust in the AS can lead to the phenomenon of automation bias, where operators disregard or do not search for contradictory information in light of an AS-generated solution which is accepted as correct.

Given that the CHAS model has been accurately tuned to a system with a sufficient amount of data, the model can be used to answer this question. Chapters 4 and 5 built confidence in the assumptions, predictive accuracy, and robustness of the CHAS model applied to the OPS-USERS system. Thus, to address this question, the initial trust level parameter was modulated to explore the predicted impact on system performance. The baseline conditions for the model were the parameter settings, described in section 4.2.1 and detailed in Appendix F, which replicated the average behavior of all test subjects in a previous experiment using the OPS-USERS testbed. Initial Human Trust is an input parameter to the CHAS model and is expressed as a percentage between 0-100%, matching the definition of Perceived Automation Capability (Section 3.4.4). The Initial Human Trust level was varied from 0 to 100% in 1% increments and the resulting impact on system performance is shown in Figure 69. The CHAS model aligns with the previous literature on trust, predicting that an operator with extremely low trust will have low performance, while an operator with extremely high trust will also have low performance. There appears to be an optimal level of initial trust around 25-50%. While 25-50% trust in the AS does

not have any practical real-world meaning, an investigation of the intervention rate and workload predictions from this same analysis is more revealing.
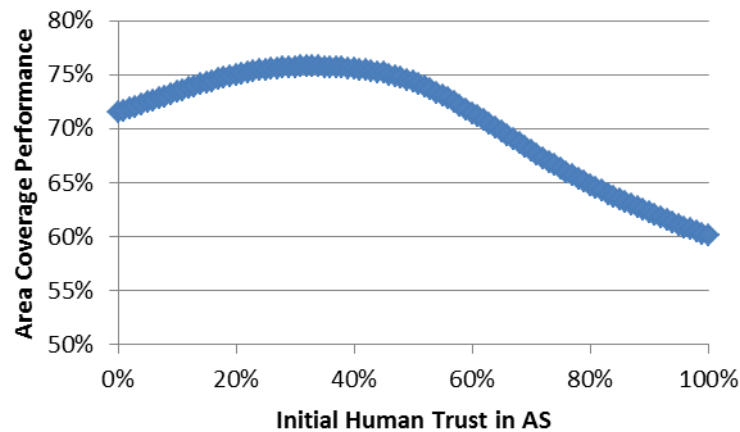
Figure 69. Model predictions of the impact of varying initial trust on system performance.

The model predicts that Initial Human Trust influences the average rate of creating search tasks and the peak utilization level of the operator as shown in Figure 70. Simulated operators with low trust intervene frequently (Figure 70a), but also have high peak workload levels (Figure 70b). The CHAS model captures that fact that at such high workload levels, operator interventions may be ineffective because the operator is cognitively overloaded. On the other hand, simulated operators with high trust rarely intervene to guide the suboptimal automation (Figure 70a). The model demonstrates the impact of this automation bias by predicting lower system performance for these overtrusting operators.
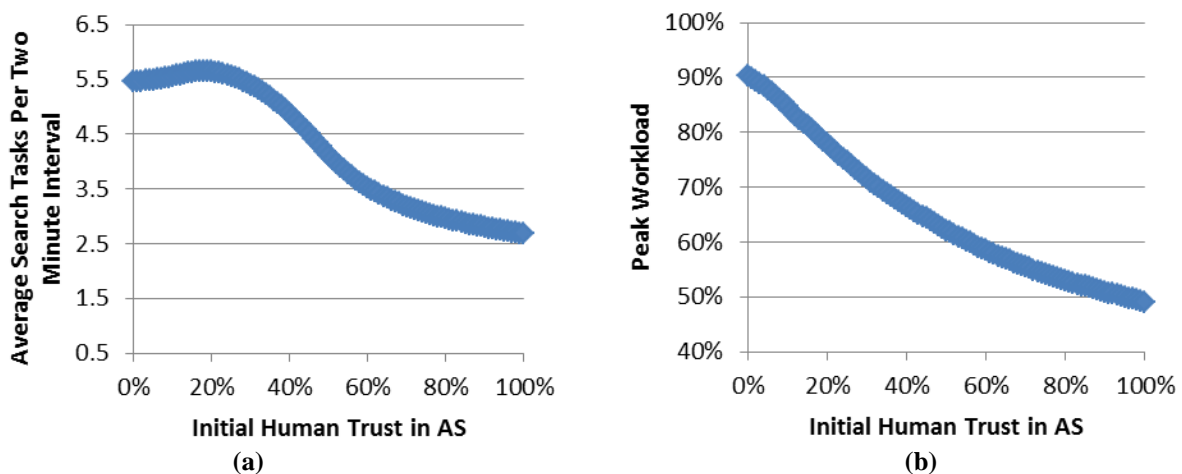
Figure 70. Model predictions of the impact of varying initial trust on (a) intervention rate and (b) workload.

203

Thus, the model predicts that the optimal level of trust, 25-50%, results in an average intervention rate between 3.5-5.5 interventions per two minute interval. This aligns with previous research on this testbed which indicated that prompting operators to intervene between every 30 and 45 seconds (~2.7-4 times per two minute interval) produced the best system performance (Clare, Maere, et al., 2012; Cummings, Clare, et al., 2010). The model makes these predictions because it assumes that operators with initial trust below 25% intervene too frequently, causing their workload to go above 70% utilization. It has been shown in the previous literature that a utilization level over 70% can lead to performance decrements (Cummings & Guerlain, 2007; Nehme, 2009; Rouse, 1983; Schmidt, 1978).

These predictions can be useful for a system designer who is considering implementing a prompting system, for example, to alert operators to intervene at a specific rate. Or for a designer who is concerned about the consequences of peak workload and how to either select or train potential operators. A major limitation of these predictions is that they are based on model assumptions about the non-linear human relationships between operator trust and the rate of interventions, the rate of interventions and human value-added to system performance, and the impact of cognitive overload on human-value added to system performance. While data was collected to estimate many of these relationships (Section 3.4), the next section explores the impact of changing these relationships on model predictions.

6.1.2 Exploring the Wider Human-System Design Space

The CHAS model enables tradespace exploration within a much larger design space that includes the human operator in addition to more typical system components included in such tradespace analyses, such as the number of UVs or the level of automation of the UVs. For example, a system designer may have different skills sets for human operators that can be selected for a specific system. Recently, the U.S. Air Force started a new program that trains UAV operators with no piloting experience (Clark, 2012). Some of these operators may be frequent gamers, who may have faster visual attentional processing (Green & Bavelier, 2003) and faster encoding of visual information into short-term memory (Wilms, et al., 2013). They may be used to multi-tasking and likely have a high tolerance for workload before cognitive overload begins. While previous research has shown that performance decrements can occur on average above 70% utilization, it is certainly possible that some operators can sustain performance up to 80%, 90%,

or even close to 100% utilization, while others begin to experience cognitive overload as low as 50% utilization. For the purposes of this section, the utilization level where cognitive overload begins will be called the cognitive overload onset point.

Thus, the CHAS model can aid a system designer in choosing the most appropriate level of intervention given the different cognitive overload onset points of various operators. The model was run with three different table functions for the Effect of Cognitive Overload on Human Value Added (Section 3.4.6, Figure 25). The baseline table function has a cognitive overload onset point of 70%, but the relationship was modified to set the cognitive overload onset point to 50% and 90%, with the results shown in Figure 71. As shown, operators who can sustain performance at utilization levels up to 90% can perhaps be prompted to intervene up to 6 times per two minute interval to maximize system performance. In contrast, operators who begin to experience cognitive overload onset at 50% utilization should only be prompted to intervene 4.5 times per two minute interval. Alternatively, operators with higher tolerances for workload can be primed to have lower trust in the AS, causing them to intervene more frequently, without fear of inducing cognitive overload. As was shown in Chapter 5, gamers who had negative *a priori* priming to lower their trust in the AS chose to intervene more frequently, but did not experience cognitive overload, enabling them to improve system performance.
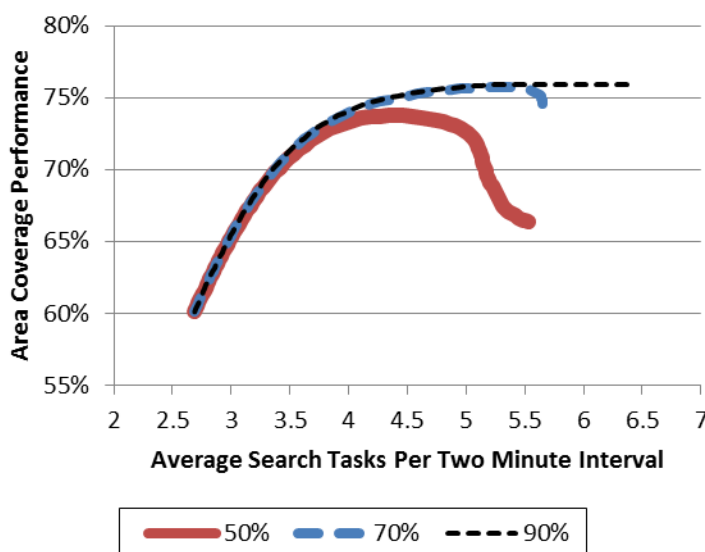
Figure 71. Intervention rate vs. system performance for different cognitive overload onset points.

It is also notable that there is not a large performance improvement when moving from 4.5 to 6 search tasks per two minute interval. This indicates that the system may be robust to varying skill levels and cognitive overload onset points, a key advantage of a goal-based architecture for real-time human-automation collaborative scheduling.

Similar limitations apply to these predictions as in the last section. However, this form of sensitivity analysis by modifying the non-linear relationships in the CHAS model can also help a system designer fine tune the model to more accurately represent the system they are analyzing.

6.1.3 Supporting Trust Calibration for Workload Specifications

Designing systems involving human operators requires understanding not just average performance, but also the range of expected performance. For example, while average workload may remain below cognitive overload conditions, it may be crucial to evaluate what percentage of operators are expected to experience significant cognitive workload at some point during a mission. System designers may want to enforce boundary conditions such as designing a system that 50% of typical operators will not exceed 70% utilization at any point during a mission.

The CHAS model can aid them in this process through Monte Carlo simulations. Using the same techniques and parameter distributions described in Section 4.3.2 and Appendix J, Monte Carlo simulations of the CHAS model were run with Initial Human Trust levels of 40%, 60%, 80%, and 100%. The results are shown in Figure 72. In order to achieve the specification of 50% of typical operators remaining below 70% utilization for the entire mission, the CHAS model predicts that operators would need to calibrate their initial trust level between 80% (Figure 72c) to 100% (Figure 72d) prior to starting the mission. This would require priming/training the operator to have complete trust in the AS prior to the mission, which should lead them to perform few interventions to guide the automation, keeping their workload at a low level.

As an alternative to calibrating trust prior to a mission, which is very difficult, system designers could investigate other methods of encouraging the appropriate rate of intervention to satisfy such workload specifications. In general, system designers could utilize similar Monte Carlo simulations to evaluate the impact of different workload specifications or design interventions on both average performance and the expected range of performance. In fact, maximizing the minimum expected performance is a common technique in robust optimization (Bertsimas &

206

Thiele, 2006), and thus the CHAS model could enable the designers of collaborative human-automation systems to adopt this technique.

Once again, one of the key benefits of the CHAS model is that it simulates the operator throughout a mission, showing the potential times when utilization may peak (early in the mission up to ~400s, as shown in Figure 72), or generally how trust, intervention rate, and performance change over time. This is in contrast to previous simulation models, which only predicted average utilization over a mission, as will be explored further in Section 6.2.
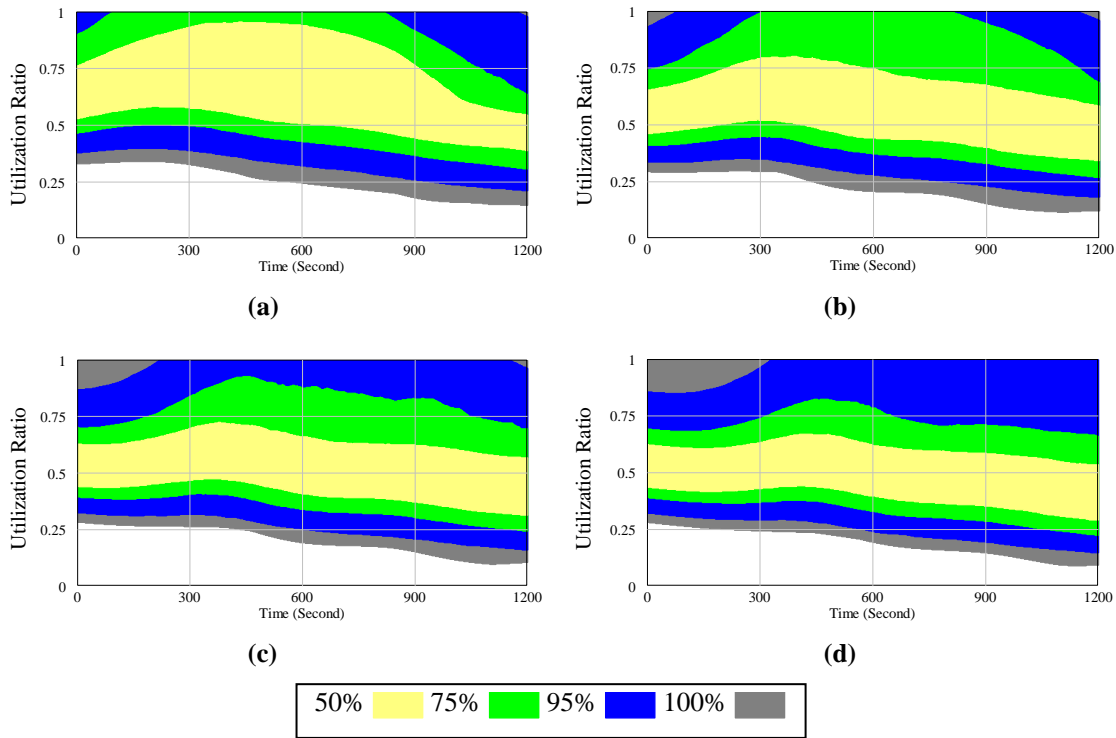
Figure 72. Monte Carlo simulations showing dynamic confidence intervals for operator utilization throughout the mission, with different initial trust levels: (a) 40%. (b) 60%. (c) 80%. (d) 100%.

6.1.4 Evaluating the Impact of Automation Characteristics

The final example of an application for the CHAS model involves evaluating the impact of automation characteristics, such as the need for certain algorithms to have time to reach consensus, on human behavior and system performance. The CHAS model was developed to model the OPS-USERS testbed (Section 3.2), a collaborative, multiple UV system which leverages decentralized algorithms for vehicle routing and task allocation (Cummings, et al., 2012). Previous research on the AS used in the OPS-USERS testbed has shown that the

automated search process is suboptimal and can be improved either with a centralized global search algorithm (Whitten, 2010) or with a collaborative human assisting the AS (Cummings, et al., 2012), both of which extend the "effective planning horizon" of the search algorithm. The *goal-based* architecture (Section 2.1) implemented in OPS-USERS was specifically designed such that if there were more tasks to do than UVs capable of completing the tasks, tasks would be left unassigned, representative of real world resource constraints. Given this architecture and analysis of a previous OPS-USERS data set (Section 3.2.2), the CHAS model assumes that an increasing rate of creating search tasks is beneficial to the performance of the system, but with diminishing returns at higher rates. It is not assumed that extremely high rates of creating search tasks has a negative impact on the automation itself, but the CHAS model does capture the potential negative impact of operator cognitive overload. This non-linear logit relationship between search tasks and system performance was quantified using data from a previous experiment (Section 3.4.5), as shown by the baseline curve in Figure 73.

In contrast, other automation architectures which rely on decentralized swarming algorithms may need time to stabilize before further operator interventions should be conducted (Walker, et al., 2012). This "neglect benevolence" concept, where high rates of intervention may actually reduce automation performance, can be represented notionally by the green dashed-line curve[2] in Figure 73. At low to moderate rates of intervention, the impact on performance is roughly the same as the baseline. However, at high rates of intervention, as opposed to diminishing returns, there is a decline in the human value added to system performance. While this curve is simply for demonstration purposes, data on the neglect time necessary for algorithm stabilization could be used to fit and validate this curve. This demonstration curve was implemented in the CHAS model through a normal distribution curve, defined by four parameters, similar to the baseline logit function (Section 3.4.5).

---

[2] Generally, System Dynamics modelers avoid implementing U-shaped curves (Section 3.4.6). Thus, the most appropriate way to model Neglect Benevolence would be through the addition of a Neglect Benevolence reinforcing feedback loop, similar to the Cognitive Overload loop. This would consist of a monotonically decreasing function showing the negative impact of high rates of intervention (which do not allow the decentralized algorithms to have enough time to reach consensus). However, for simplicity, this curve was implemented using a U-shaped curve purely for demonstration purposes.
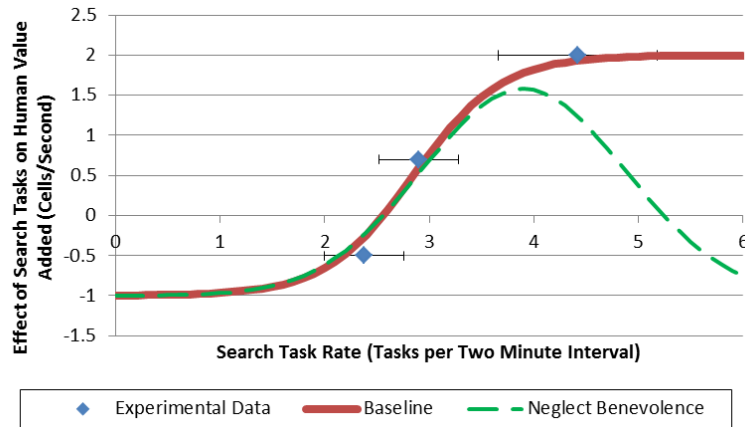
Figure 73. Two potential relationships between Search Task Rate and Effect of Search Tasks on Human Value Added. Empirical data shown with ±1 Standard Error bars.

Similar to Section 6.1.1, Initial Human Trust was varied from 0 to 100% and the impact on system performance is shown in Figure 74. Recall that the CHAS model assumes that a high level of human trust should result in a low rate of intervention. Thus, at high levels of trust, the performance of the baseline automation architecture as compared to the Neglect Benevolence architecture is the same. However, for operators with lower levels of trust, below 50%, the CHAS model predicts that there will be a sharp decrease in system performance. The CHAS model predicts that these operators would intervene by creating 5.25 search tasks per two minute interval. According to Neglect Benevolence curve in Figure 73, this rate of intervention would actually generate negative Human Value Added, as the operator is negatively impacting the performance of the automation. Data analysis in Section 5.7.1 supports this concept, showing that some operators can in fact hurt system performance.

It should be noted that system performance does not continue to decrease below 30% Initial Trust for the Neglect Benevolence curve in Figure 74. This is caused by the model's representation of the impact of cognitive overload. All operators who begin with initial trust below 30% are predicted to have an average utilization level near or above 70%. The model assumes that interventions will become less impactful on system performance once the operator reaches cognitive overload, as high levels of stress have been shown to induce perceptual narrowing (Kahneman, 1973). This assumption may hold when frequent operator interventions cannot directly harm automation performance. However, for automation architectures that need time to stabilize before further operator interventions should be conducted, this assumption does

not hold. The model does not capture the fact that even ineffective interventions under conditions of cognitive overload can continue to negatively impact decentralized algorithms. Performance should continue to drop with lower levels of trust below 30% that cause higher rates of intervention. Refining the implementation of the cognitive overload loop to capture this fact is discussed further in the future work section of Chapter 7.
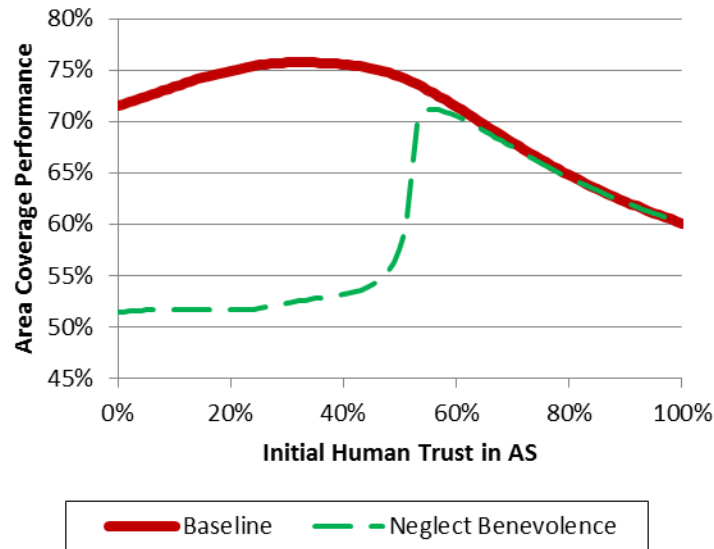
Figure 74. Model predictions of the impact of varying initial trust on system performance given different automation characteristics.

In summary, four example use cases of the CHAS model have been shown to illustrate how it could aid UV system designers. Next, the CHAS model's accuracy and features are compared with a previously developed Discrete Event Simulation (DES) model of human supervisory control of multiple UVs.

6.2 Comparison of SD and DES Model

One of the major goals of this thesis was to attempt to adapt System Dynamics modeling techniques to model human-automation collaboration for scheduling multiple semi-autonomous UVs. One of the research questions posed in Chapter 1 was "How does it compare to other relevant models?" Thus, it is important to compare both the quantitative accuracy as well as the qualitative features of the CHAS model to a previously developed computational model of human supervisory control of multiple UVs.

Nehme (2009) developed a queuing-based, multi-UV discrete event simulation (MUV-DES) model of human supervisory control of multiple UVs. MUV-DES captures both UV variables, such as the types of UVs and the level of autonomy of the vehicles, as well as operator variables, such as attention allocation strategies and Situation Awareness (SA). Similar to the CHAS model, MUV-DES aims to support the designers of future UV systems by simulating the impact of alternate designs on vehicle, operator, and system performance. MUV-DES uses DES/queuing-based constructs including events, arrival processes, service processes, and queuing policies to model the human operator as a serial processor of tasks. The input variables to MUV-DES are primarily the distributions of the arrival rate of various operator tasks and the distributions of service times for these tasks. These distributions are drawn from previous experimental data.

While there is an ongoing argument in the modeling community about the scenarios for which SD is more appropriate than DES (i.e. (Özgün & Barlas, 2009; Sweetser, 1999)) a key contribution of this thesis is the evaluation of the adaptation of SD techniques to model human supervisory control of UVs in comparison to DES techniques. To begin this comparison, both MUV-DES and CHAS were used to simulate the high task load OPS-USERS experiment (Section 4.2.2). In this experiment, operators were prompted to view automation-generated schedules at prescribed intervals of either 30 or 45 seconds. Changing the rate of prompts to view new schedules modulates the task load of the operator, such that 30s replan intervals should induce higher workload than the 45s intervals. The average utilization of operators in the 30 and 45s replan prompting interval conditions is shown in Figure 75. The CHAS model was applied to simulate this experiment (Section 4.2.2) while distributions of arrival rates and service times were generated from experimental data for use in MUV-DES model. The utilization results from the CHAS and MUV-DES simulations are compared to the experimental results in Figure 75.

The MUV-DES model was more accurate for the 30 second replan prompting interval condition as compared to the CHAS model, which had an average utilization prediction that was higher than the experimental data. For the 45 second replan prompting interval data, both models were slightly off, although both predictions fell within the 95% confidence interval of the data. Both models captured the decrease in utilization from the 30 to 45 second replan prompting interval conditions.
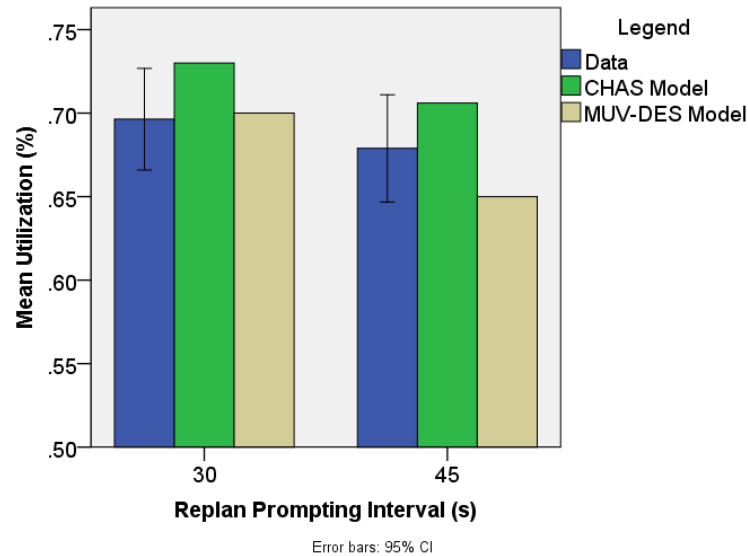
Figure 75. Average utilization for OPS-USERS high task load experiment, with CHAS and MUV-DES predictions. 95% confidence intervals for the experimental data are shown.

While both models replicate average utilization values for the experimental data, a qualitative comparison of the features of MUV-DES and the CHAS models is more revealing. There are certainly some similarities, such as capturing the impact of delays in operator perception on SA and modeling the impact of high operator workload on system performance. However, Nehme wrote that the MUV-DES model lacked the "consideration of interaction between human-UV system variables (i.e. the variables that form inputs to the model) [which] can result in lower predictive accuracy" (Nehme, 2009, p. 152). In addition, Nehme wrote that "the research in this thesis focused on situational awareness. However, there are other operator characteristics which can significantly influence UV-team performance, such as operator trust" (Nehme, 2009, p. 153).

The CHAS model aimed to address these limitations of the MUV-DES model. There are three major features of the CHAS model that build upon the MUV-DES model:

- The CHAS model captures the feedback interactions among perception, workload, trust, decisions to intervene, and performance. Rather than treating the aforementioned variables as separate factors, the CHAS model captures the interaction among these components.

- The CHAS model explicitly represents qualitative variables such as human trust and its impact on the rate at which humans intervene into the operations of the team of UVs, and thus on system performance. The dynamics of trust are captured by enabling trust to adjust over time throughout the mission with some inertia.

212

- The CHAS model can provide predictions of continuous measures, a key attribute of all SD models. While the MUV-DES model provided predictions of system performance based on the occurrence of events (i.e. visually identifying targets), this form of performance prediction is not as useful for a continuous performance metric such as area coverage.

Both the MUV-DES model and the CHAS model have specific domains for which they are most appropriate. The MUV-DES model is best suited for using probabilistic distributions to accurately model an operator who is a serial processor of discrete tasks, such as visually identifying targets. The CHAS model is better suited for modeling continuous performance feedback that is temporally dependent and capturing the impact of qualitative variables such as trust. Based on this, the CHAS model enables diagnosticity, allowing a system designer to more precisely characterize the reasons behind behavior and performance patterns.

6.3 Model Generalizability

In addition to evaluating the adaptation of SD modeling techniques, one of the aims of this research was to create a model that was generalizable to a variety of scenarios where a human operator collaborates with an automated scheduler for real-time scheduling of multiple vehicles in a dynamic, uncertain environment. These scenarios could include searching a designated area, finding and tracking moving targets, visiting designated locations for delivery or pickup, etc.

The CHAS model was developed through an inductive modeling process. Größler and Milling defined the inductive SD modeling process by stating that "the solution to a specific problem is sought as well as a specific situation serves as the basis for the model. Later in the process, insights gained in the project might be generalized…" (Größler & Milling, 2007, p. 2). The process began by analyzing a specific experimental data set in order to develop a dynamic hypothesis, defined as a theory that explains the behavior of the system as an endogenous consequence of the feedback structure of the holistic system (Sterman, 2000). The dynamic hypothesis of human-automation collaborative scheduling which was developed was: if operators can either a) anchor to the appropriate trust in the automation and expectations of performance earlier in the mission and/or b) adjust their trust and expectations faster through better feedback about the automation, then system performance should improve.

This hypothesis, which has been validated throughout the development and testing of the CHAS model on a variety of data sets, is in fact generalizable to wider variety of scheduling domains. Regardless of whether an operator is collaborating with an AS for real-time multi-UV control, Air Traffic Control (ATC) planning, rail operations scheduling, manufacturing plant operations, or space satellite control, appropriate trust and better feedback about the automation should lead to superior performance. Many of these domains require scheduling under significant uncertainty, which means that human judgment and adaptability must be leveraged in order to guide automation in a collaborative scheduling process. For example, Section 4.3.3 demonstrated how the CHAS model could be generalized to a different domain, an Urban Search and Rescue (USAR) mission with a large team of robots.

However, the CHAS model assumes that the operator is working in a *goal-based*, decentralized control architecture, where each vehicle has onboard autonomy enabling it to compute its locally best plan to accomplish the mission goals with shared information. Many of the other domains mentioned above still rely on completely centralized scheduling methods, where information is collected from each vehicle, an attempt is made to develop a globally optimal schedule, and then the plan is sent out to all vehicles. This is a limitation of the generalizability of the CHAS model, especially in the computation of workload, as the operator would need to dedicate far more mental resources to scheduling in a centralized architecture. However, these domains will likely move towards *goal-based*, decentralized control architectures in the future. This would enable systems to respond to changes in the environment more quickly, scale to larger numbers of vehicles while taking advantage of each vehicle's added computational power, and maintain performance despite communications failures (Alighanbari & How, 2006; Whitten, 2010).

To further demonstrate how the model could be generalized for use in these other scheduling domains, the model terminology has been adjusted to more general terms, as shown in Figure 76. Common among the many possible scenarios in which a human operator is collaborating in real-time with an automated scheduler is the accomplishment of mission goals. Thus, the system performance module has been generalized to focus on the completion of mission goals. These mission goals could include a measure of how much area has been covered for a search mission, a measure of how many victims have been found in a search and rescue mission, or how many locations have been visited for a delivery mission.

Trust

Trust Change
Time Constant

Initial Human
Trust

Human
Trust

Change in
Trust

+

Perceived
Automation
Capability

Base PPG

PPG Max Slope

B1

Trust in
Automation

-

Interventions

Min Intervention
Rate

Max Intervention
Rate

Midpoint Trust

Trust Max Slope

-

Intervention
Rate

Base Intervention Rate

Intervention Rate
Max Slope

Min Human Value Added

Max Human Value Added

+

Effect of Interventions
on Human Value Added

Workload

Average Intervention
Length of Time

+ +

Human
Workload

+ -

Initial NST

Nonscheduling
Taskload (NST)

NST Rate

R2

Cognitive
Overload

-

Effect of Cognitive
Overload on Human
Value Added

Table for Cognitive
Overload

Perception of Performance

Initial EP

Expected
Performance
(EP)

Change in
EP

Time Horizon for
Expected
Performance

+

Perceived
Performance Gap
(PPG)

+

R1

Expectations
Adjustment

-

Perceived Present
Performance (PPP)

+

Time to Perceive
Present Performance

System Performance

Total Mission
Goals

Automation
Generated
Performance

Human Value
Added

+

+

Mission Goals
Incomplete

+ +

System
Performance

+

Mission Goals
Completed

+

Percent of Goals
Completed

-

Percent of Goals
Remaining

<Total Mission
Goals>

-

+

Figure 76. Generalized version of the CHAS model.

215
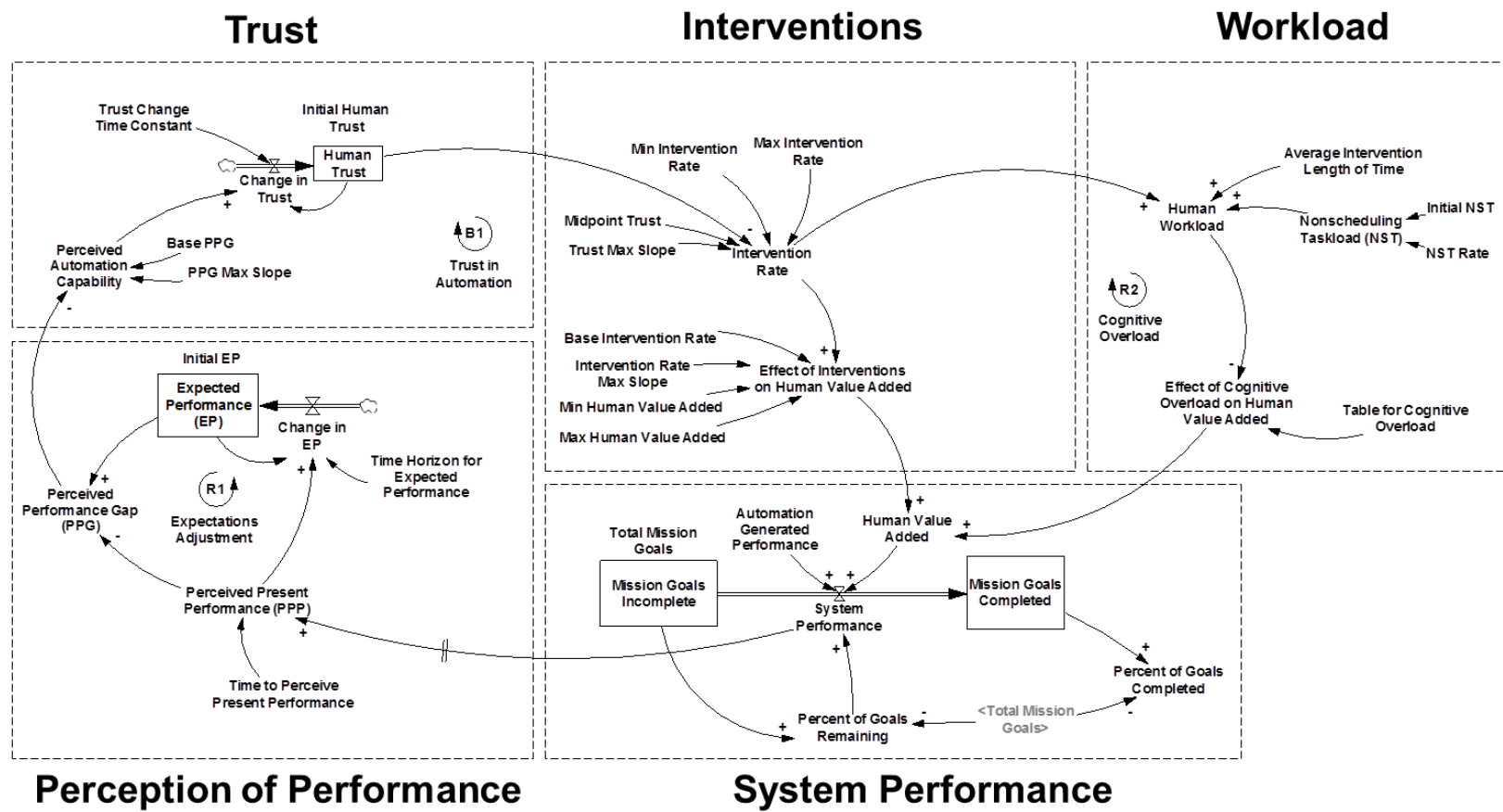
In order to model the system, it is still assumed that there are a total number of mission goals (sometimes, but not always known to the human operator or automation). For an area coverage and target finding mission, for example, there might be a defined limit to the total area to cover. For a mission without a clearly defined boundary or end-point, the Total Mission Goals parameter can be set to a near-infinite number to represent an ongoing mission. System Performance is measured by the rate at which mission goals are accomplished, for example the rate of finding new victims in a search and rescue operation. Finally, it is still true that once there are few incomplete mission goals left to accomplish, such as finding the last victim in a search and rescue scenario, the rate of accomplishing mission goals, System Performance, must decrease.

The method by which humans add value depends on the specific system being modeled (see Section 2.2.6 for further discussion). For example, naval aircraft carrier deck operators apply heuristics to quickly change the schedules for launch catapults and a variety of aircraft while minimizing disruptions (Ryan, 2011; Ryan et al., 2011). Operators can change the objective function of an algorithm while planning for a multi-UAV mission (Malasky, et al., 2005) or decide when to loosen constraints on the scheduling problem for a network of satellites (Howe, et al., 2000). Also, operators can manually modify AS-generated solutions, guide the algorithm to replan for only a small portion of the problem, or backtrack to previous solutions for a multiple vehicle delivery scheduling problem (Scott, et al., 2002). Finally, the generalized CHAS model still assumes that the rate of operator interventions to guide the suboptimal automation is negatively non-linearly dependent on Human Trust level.

One of the major assumptions that still remains in the generalized model is that the automation which the operator is guiding is suboptimal. Some scheduling domains may be well-defined enough for automation to provide near-optimal solutions, and thus human intervention is unnecessary. However, many scheduling problems, such as assigning multiple vehicles or satellites to many possible tasks with capability, location, and timing constraints in uncertain, dynamic environments are likely situations with unbounded indeterminacy (Russell & Norvig, 2003). The set of possible preconditions or effects either is unknown or is too large to be enumerated completely. It is thus likely that the optimization problem is NP-hard, meaning that the AS cannot find an optimal solution in polynomial time (Russell & Norvig, 2003). Also, the

objective function for this optimization may be non-convex, meaning that certain algorithms may become stuck in local minima. In many of these cases, where either the environment is stochastic or the search space is large, it is unlikely that an optimal plan can be found or will remain optimal throughout the mission. In addition, the definition of "optimal" may be difficult to quantify and represent in a single, static objective function for a command and control situation, where unanticipated events such as weather changes, vehicle failures, unexpected target movements, and new mission objectives often occur. Thus, in the presence of unknown variables, possibly inaccurate information, and changing environments, automated scheduling algorithms can be "brittle" and can benefit from human guidance.

In summary, the CHAS model can be applied to a variety of system design problems across a number of domains that involve real-time human-automation collaborative scheduling.

6.4 Model Limitations

Finally, one of the research questions posed in Chapter 1 was "What are the boundary conditions of the model?" Based on the results of the multi-stage validation process, the predictive validation experiment, and the potential applications of the CHAS model presented in this Chapter, a number of limitations have been identified. These limitations can be categorized into three categories: computational modeling assumptions and simplifications, the need for data to tune the model, and model boundary conditions.

First, the CHAS model is a computational model of human behavior and decision-making. In contrast to purely descriptive or conceptual models, computational models typically leverage computer simulations to both promote deeper understanding of how human operators behave and provide testable predictions about human behavior under different circumstances (Gao & Lee, 2006; Parasuraman, 2000). While there are numerous benefits to developing and using a computational model, all such models make a number of difficult-to-test assumptions about the complexities of human perception, cognition, emotions, and decision-making. Reducing the judgment, adaptability, flexibility, and reasoning abilities of humans to a few formulae is certainly a simplification and such models will never fully capture the variability from individual to individual human. Given these limitations, however, the CHAS model can still be useful to a system designer, as described earlier in this Chapter. While the model makes certain inferences

about perceptions, expectations, and trust, these variables are secondary to the more quantifiable metrics, such as the frequency of interventions, the percent busy time during a mission, or overall system performance. Capturing the interactions and relationships between measureable human behavior and the underlying characteristics of human operators is an ongoing and important endeavor, which always has room for improvement.

The second major limitation of the CHAS model is the need for sufficient data to validate and tune the causal relationships throughout the model. While the three historical data sets and new data set presented in this thesis were utilized in an attempt to validate as many model assumptions as possible, further work is necessary to validate some of the causal relationships in the model. For example, future work should gather additional data to characterize in more detail the negative, non-linear relationship between PPG, Perceived Automation Capability, and Human Trust. Also, it must be acknowledged that these four data sets all come from simulation-based experiments with MIT students serving as test subjects. Thus, it remains an open question how these results would generalize to the broader population and non-simulation environments where actual, expensive UVs and potentially human lives are on the line. Finally, the need for prior data to tune the model means that the CHAS model is limited to assisting designers who aim to make evolutionary changes to existing systems. It would be desirable to assist designers of revolutionary systems, those that are radically different from previous systems. However, without some data about human interactions with the automation, such a model would not be accurate enough to be useful.

It should be noted that the CHAS model was specifically tuned for the OPS-USERS testbed (Figure 12), and utilizes area coverage as the primary performance metric. As explained in more detail in Section 3.2, the area coverage metric provided beneficial features for developing and testing a System Dynamics model, as it is a continuous measure of performance. However, in a typical OPS-USERS mission, the goal is not simply to maximize area coverage, but to find and track moving ground targets. Thus, performance metrics such as targets found are crucial to the validity and usefulness of the CHAS model. Earlier versions of the CHAS model (Appendix A) estimated the number of targets found based on an empirical, logarithmic relationship between area covered and targets found. This relationship had an R^2 value of 0.60, meaning that area coverage could explain 60% of the variance in targets found. While the relationship was

218

removed because it did not directly contribute to a feedback loop in the final, parsimonious CHAS model, it would be simple to add this relationship back to the model. Additionally, future work should investigate the addition of new model constructs to more accurately calculate targets found and other important performance metrics.

The third major limitation of the CHAS model relates to the boundary conditions of the model. As described in Section 4.1.1, the "model boundary" (Sterman, 2000), or the scope of the CHAS model was chosen to keep the initial model simple enough to attempt to validate given the available experimental data. The CHAS model in its current form is limited to simulating a single operator in moderate to high task load missions, given the real-time human-automation collaborative scheduling system presented in Section 2.1. This definition includes a *goal-based* architecture (Clare & Cummings, 2011), where the vehicles are semi-autonomous and with the guidance of the AS, can conduct much of the mission on their own. The human operator only guides the high-level goals of the vehicles, as opposed to guiding each individual vehicle. This means that the human operator is monitoring the system and makes decisions to intervene throughout the mission in order to adjust the allocation of resources at a high level. The CHAS model would not be well-suited for modeling lower-level, manual control of robotic vehicles. Also, the CHAS model assumes that the operator is physically removed from the environment that the UVs are operating in and conducts supervisory control through a computer interface. In its current form, the model is not suited for modeling manned-unmanned teaming or human-robot personal interaction.

An additional boundary condition of the model is the assumption that the operator begins the mission with sufficient Situation Awareness (SA) and has a general understanding of what he or she is viewing on the interface, where the UVs are, which targets require tracking, etc. For the experimental datasets used to validate the model, all operators first went through a tutorial explaining the interface and a practice mission to become familiar with the system. Few order effects were found for the actual experiment missions (Appendix T), thus it appears that this assumption was an acceptable simplification. Future work could add to the model to capture the initial transients in operator perceptions, behavior, and workload prior to achieving SA. Similarly, there are often differences between novices and experts in their behavior and interactions with the system. As shown in Chapter 5, operators who played video games

frequently had higher initial trust in the AS and were able to click around the interface faster, accomplishing the same tasks in less time and making faster decisions. While the model attempts to capture these differences through a set of human and system parameters, such as Initial Human Trust and Length of Time to Replan, future work could further study the differences between novices and experts and expand the model constructs to better capture these differences.

Finally, the CHAS model could potentially be expanded to include many other features. For example, consideration of teams of operators controlling teams of UVs through the addition of multiple feedback interactions. This would include modeling team communication, the impact of team structure on interaction with the automation, as well as interpersonal trust in addition to simply trust in the automation. Also, the CHAS model could be extended to take into account the impact of low task load, vigilance missions. All of the data used to build and test the CHAS model came from medium to high task load missions. The CHAS model would need to be tested on low task load experimental data and refined as necessary to take into account the negative impact of vigilance on human performance. Finally, the implementation of multi-UV systems in real-world settings will require an intensive investigation of how both operators and automation respond to vehicle failures, communications delays, and safety policies. All of these attributes could be added to a future iteration of the CHAS model.

6.5 Summary

This chapter presented four example use cases of the CHAS model, to illustrate how it could aid UV system designers. First, it was shown that designers could use the model to further investigate the impact of operator trust in the Automated Scheduler (AS) on system performance. Second, the CHAS model was used to explore a wider system design space that includes both traditional system components as well as human characteristics, such as cognitive overload onset points. Third, it was shown that through Monte Carlo simulations, the CHAS model could support robust system design to specifications such as maximum workload levels. Finally, designers can evaluate the impact of different automation characteristics, such as the need for certain algorithms to have time to reach consensus, on human behavior and system performance.

Then, the CHAS model's accuracy and features were compared with a previously developed Discrete Event Simulation (DES) model of human supervisory control of multiple UVs. It was

shown that both models replicate average utilization values for the experimental data. The CHAS model has three major features that build upon the MUV-DES model. First, the CHAS model captures feedback interactions among components. Second, the CHAS model explicitly represents qualitative variables such as human trust and its impact on the rate at which humans intervene into the operations of the team of UVs, and thus on system performance. Third, the CHAS model can provide predictions of continuous measures, a key attribute of all SD models. Both the MUV-DES model and the CHAS model have specific domains for which they are most appropriate. The MUV-DES model is best suited for using probabilistic distributions to accurately model an operator who is a serial processor of discrete tasks, such as visually identifying targets. The CHAS model is better suited for modeling continuous performance feedback that is temporally dependent and capturing the impact of qualitative variables such as trust. Based on this, the CHAS model has improved diagnosticity, enabling a system designer to more precisely characterize the reasons behind behavior and performance patterns.

Finally, the generalizability of the model was discussed along with model limitations. The CHAS model could be generalized to a variety of scenarios where a human operator is collaborating with an automated scheduler in a *goal-based* architecture for real-time scheduling of multiple vehicles in a dynamic, uncertain environment. These scenarios could include reconnaissance missions, package delivery, multi-UV control, Air Traffic Control (ATC) planning, rail operations scheduling, manufacturing plant operations, or space satellite control. Many of these domains require scheduling under significant uncertainty, which means that human judgment and adaptability must be leveraged in order to guide automation in a collaborative scheduling process. Thus, appropriate trust and better feedback about the automation should lead to superior performance. In addition, a number of limitations of the CHAS model have been identified through the multi-stage validation process, the predictive validation experiment, and the potential uses of the CHAS model presented in this Chapter. These limitations were categorized into three categories: computational modeling assumptions and simplifications, the need for data to tune the model, and model boundary conditions.

7 Conclusions

Recent advances in autonomy have enabled a future vision of single operator control of multiple heterogeneous Unmanned Vehicles (UVs). Real-time scheduling for multiple UVs in uncertain environments will require the computational ability of optimization algorithms combined with the judgment and adaptability of human supervisors. Automated Schedulers (AS), while faster and more accurate than humans at complex computation, are notoriously "brittle" in that they can only take into account those quantifiable variables, parameters, objectives, and constraints identified in the design stages that were deemed to be critical. Previous experiments have shown that when human operators collaborate with AS in real-time operations, inappropriate levels of operator trust, high operator workload, and a lack of goal alignment between the operator and AS can cause lower system performance and costly or deadly errors. Currently, designers trying to address these issues test different system components, training methods, and interaction modalities through costly human-in-the-loop testing. The Collaborative Human-Automation Scheduling (CHAS) model was developed to aid a designer of future UV systems by simulating the impact of changes in system design and operator training on human and system performance. This chapter summarizes the important results in the CHAS model's development and validation. Also, this chapter evaluates how well the research objectives were met, suggests potential future work, and presents the key contributions of this thesis.

7.1 Modeling Human-Automation Collaborative Scheduling

Real-time human-automation collaborative scheduling of multiple UVs was defined as a potential future method for a single human operator to control multiple heterogeneous UVs (air, land, sea) by guiding an Automated Scheduler (AS) in a collaborative process to create, modify, and approve schedules for the team of UVs, which are then carried out by the semi-autonomous UVs. The representative setting for this thesis was a reconnaissance mission to search for an unknown number of mobile targets. The mission scenario was multi-objective, and included finding as many targets as possible, tracking already-found targets, and neutralizing all hostile targets. Scheduling was defined in this thesis as creating a temporal plan that assigns tasks/targets among the team of heterogeneous UVs, determines when the tasks will be completed, and takes into account capability, location, and timing constraints. In order to

conduct this mission in uncertain, dynamic environments, a human operator would need to collaborate with a decentralized AS through a *goal-based* architecture to guide the team of UVs.

Based on this definition, a review of previously developed relevant models of humans in scheduling and control situations was conducted to evaluate their applicability to the representative setting. The review identified crucial gaps with regards to real-time human-automation collaborative scheduling of multiple UVs. Previous computational models did not capture the feedback interactions among important aspects of human-automation collaboration, the impact of AS characteristics on human behavior and system performance, and the impact of qualitative variables such as trust in automated scheduling algorithms.

7.1.1 CHAS Model

Through a review of the relevant literature, six attributes that were important to consider when modeling real-time human-automation collaborative scheduling were identified, providing a theoretical basis for the model proposed in this thesis. These attributes were attention allocation and situation awareness, cognitive workload, trust in automation, human learning, automation characteristics, and human value-added through interventions.

A computational model was then developed that incorporated all of these attributes. Following the System Dynamics (SD) modeling process, a previous experimental data set was analyzed, which led to the creation of a dynamic hypothesis: if operators can either a) anchor to the appropriate trust in the AS and expectations of performance earlier in the mission and/or b) adjust their trust and expectations faster through better feedback about the AS, then system performance should improve. Using this dynamic hypothesis, the data analysis, and prior literature in human supervisory control, three major feedback loops were developed. These feedback loops were implemented into a SD simulation model, the Collaborative Human-Automation Scheduling (CHAS) model. The CHAS model utilizes SD constructs (stocks, flows, causal loops, time delays, feedback interactions) to model real-time human-automation collaborative scheduling of multiple UVs.

7.1.1.1 Model Inputs and Outputs

The exogenous input parameters to the CHAS model, which a system designer could use to explore the design space, consist of four categories:

- *System characteristics*: Operators engaged in these dynamic, high workload environments must both concentrate attention on the primary task (e.g., monitoring vehicle progress and identifying targets) and also be prepared for various alerts, including incoming chat messages or automation notifications about potential changes to the vehicle schedules. Thus, the first category of input variables includes the Nonscheduling Task Load (NST) that an operator must deal with, as compared to the scheduling activities that the operator chooses to conduct. Also, the total amount of area to be searched during the reconnaissance mission is considered a system characteristic.

- *Automation characteristics*: The model explicitly represents the contribution of the automation to the performance of the system. This provides the system designer with the ability to quantitatively evaluate the impact of improved automation, such as a better search algorithm, on the performance of the system and potentially on the operator's trust level and workload. Also, the model uses an exogenously defined, non-linear relationship to capture the effect of operator interventions on the automation and thus on system performance. As described in Chapter 6, this could potentially allow a system designer to investigate the impact of evolutionary changes to a current system, such as using a different AS, on human and system performance.

- *Human-automation interaction time lengths and frequencies*: The length of time that an intervention takes is modeled as an input parameter. Interventions by the human operator can include replanning to request a new schedule from the AS or creating a search task to guide the search pattern of the team of UVs. If the designer is able to reduce the time that these interactions take or influence the frequency of these interventions, the impact on operator workload and performance can be evaluated.

- *Human characteristics*: The initial conditions of the human operator (i.e. initial trust level) are modeled as input parameters. In addition, the CHAS model assumes that a number of human characteristics and time constants (i.e. the time constant for adjusting trust) are static and could be modeled as exogenous parameters. These parameters can capture a variety of

human attributes, including the impact of previous experiences with automation and video games. Finally, a set of non-linear relationships, such as the impact of cognitive workload on human value added to system performance, were estimated from experimental data where possible and are set by exogenous input parameters or table functions. Chapter 5 demonstrated the impact of changing the initial trust level of human operators on performance. Also, as described in Chapter 6, by varying these non-linear relationships, the impact of different cognitive overload onset points on performance can be modeled.

All of these inputs could also be studied as outputs. In addition, the CHAS model provides a number of endogenously-calculated variables that a designer might be interested in capturing. A designer could investigate system performance by analyzing the rate of area coverage or the total area coverage by the end of the mission. Changes in human trust in the AS can be captured, which can be beneficial for the designer to understand, as both undertrust (Clare, Macbeth, et al., 2012) and overtrust (Parasuraman & Riley, 1997) in automation have been shown to hurt the performance of a system. The rate at which a human operator decides to intervene can be analyzed. This can be important for a system designer to know, for example, to analyze the impact of communications delays between the operator and the vehicles in a decentralized network of UVs (Southern, 2010). The effect of alternative system designs on the workload of the operator can also be captured. Since operator workload is a key driver of human performance, the ability to design a future UV system with an understanding of the impact on operator workload is crucial.

7.1.1.2 Model Benefits

The CHAS model addresses three gaps with regards to real-time human-automation collaborative scheduling of multiple UVs that were identified among the previously developed models reviewed in Chapter 2. First, the CHAS model captures the feedback relationships among perception, workload, trust, decision-making, and performance. Components are not static in the CHAS model, but can change over time throughout the simulation of a mission due to feedback interactions. Second, the CHAS model captures crucial details of the automation used in a real-time human-automation scheduling system. These details include the contribution of the automation to the performance of the system and the impact of operator interventions to guide the automation on system performance. This provides a system designer with the ability to

226

quantitatively evaluate the impact of different algorithms and interaction modalities on system performance and operator trust and workload. Third, the model explicitly represents human trust and its impact on the rate at which humans intervene into the operations of the team of UVs. The dynamics of trust are captured by enabling trust to adjust over time throughout the mission with some inertia. The alignment of human and AS goals influences both their expectations of how well the system should perform and their perception of the capability of the AS, both of which are captured in the model.

By addressing the three gaps above, the CHAS model built upon a previously developed Discrete Event Simulation (DES) model of human supervisory control of multiple UVs (Nehme, 2009). In Chapter 6, it was shown that both models can successfully replicate average utilization values for experimental data. However, the DES model and the CHAS model have specific domains for which they are most appropriate. The DES model is best suited for using probabilistic distributions to accurately model an operator who is a serial processor of discrete tasks, such as visually identifying targets. The CHAS model is better suited for modeling continuous performance feedback that is temporally dependent and capturing the impact of qualitative variables such as trust. Based on this, the CHAS model enables diagnosticity, allowing a system designer to more precisely characterize the reasons behind behavior and performance patterns.

The CHAS model can aid a designer of future UV systems in predicting the impact of changes in system design and operator training. This can reduce the need for costly and time-consuming human-in-the-loop testing that is typically required to evaluate such changes. It can also allow the designer to explore a wider trade space of system changes than is possible through prototyping or experimentation. The CHAS model can be used to investigate the impact of trust level on performance, explore the design space for operators with different levels of cognitive overload onset, aid in designing systems to workload specifications, and investigate the impact of neglect benevolence time for decentralized algorithms to stabilize on human and system performance. These four potential applications are discussed further in Section 7.2.

7.1.2 Model Confidence

Through a multi-stage validation process, the CHAS model was tested on three experimental data sets to build confidence in the accuracy and robustness of the model under different conditions. Additionally, a new human subject experiment was conducted to evaluate the ability of the CHAS model to predict the impact of changes in system design and operator training on human and system performance. The experiment had three independent variables: priming to influence initial operator trust, a system design change to prime expectations of system performance, and a time delay in feedback on system performance. The experiment gathered near real-time data on operator perceptions of performance, expectations of how well the system should be performing, and trust in the AS. All three of these variables were essential to the CHAS model and collecting this data enabled the evaluation of model assumptions. While no model can ever be truly validated, results from both the replication and predictive validation testing have built confidence in the CHAS model's accuracy and robustness, which are discussed in more detail below.

7.1.2.1 Model Accuracy

First, the CHAS model was able to replicate the results of the real-time human-automation collaborative scheduling experiment that provided data to inform construction of the model. The model was able to capture the differences in system performance and rates of intervention between high and low performers. The model was also able to accurately simulate changes in output variables over time, including the decline in workload throughout the mission. Additionally, the model replicated oscillations in intervention rate which were seen in the data set, as operators sought the appropriate level of trust and the rate of intervention that would produce performance that matched their expectations.

Second, a slightly modified version of the CHAS model was able to replicate the results of a second experiment in which operators were subjected to a much higher task load. The model replicated the impact of different replan prompting intervals accurately by showing an increase in workload, a change in search task intervention rates, and no significant difference in performance between the two groups. The model's predictions of system performance were accurate because the model simulated the negative impact of cognitive overload. This built

228

confidence in the model's method of capturing the impact of cognitive overload on human decision-making and system performance.

Third, as an external validation, a tailored version of the CHAS model was also used to replicate a data set from a multi-robot Urban Search and Rescue (USAR) experiment. The only major change to model structure was a simplification of the workload calculation. The model accurately captured the impact of an increased rate of interventions on system performance. The long-term learning behavior of operators in the data set was accurately replicated, as operators adjusted their expectations of performance and trust in the automation over time. Once again, the model captured the diminishing returns of extremely high rates of interventions due to cognitive overload. This external validation test demonstrated the ability to generalize the model for use with other real-time human-automation collaborative scheduling systems.

Fourth, the predictive accuracy of the model was tested through a new human subject experiment. Test subjects who play computer and video games frequently were found to have a higher propensity to over-trust automation. By priming these gamers to lower their initial trust to a more appropriate level, system performance was improved by 10% as compared to gamers who were primed to have higher trust in the AS. The CHAS model provided accurate quantitative predictions of the impact of priming operator trust on system performance. Also, system design changes were implemented in the experiment to influence operator expectations and perceptions of performance. When provided with accurate input data on operator responses to these system design changes, the CHAS model made accurate predictions of the impact on system performance.

Both quantitative and qualitative data from this experiment validated a number of the assumptions in the CHAS model. Operators adjusted their expectations of performance over time, supporting the need for an expectations adjustment feedback loop in the CHAS model. Results showed that as operators perceived that the AS was performing below their expectations, they lost trust in the AS. Finally, experimental data supported the overall dynamic hypothesis of the CHAS model, showing that high performers anchored to the appropriate level of trust early in the mission, understanding the imperfections in the automation and compensating to improve system performance.

7.1.2.1 Model Robustness

In addition to testing the model's ability to accurately replicate experimental data, the validation tests presented in Chapter 4 also evaluated the model's robustness under a variety of conditions.

First, the adequacy of the model boundary was evaluated by comparing the endogenous and exogenous variables in the model to determine whether the model was appropriate for the purpose for which it was built and whether the model included all relevant structure. Also, tests were conducted to show that the model structure captures important aspects of real-time human-automation collaborative scheduling, such as the tension between the positive impact of operator interventions and the effectiveness of those interventions once the operator is cognitively overloaded.

Second, a set of extreme conditions tests were conducted and in all cases, the model behaved as expected. Through the use of logit curves to characterize key causal relationships, the model was robust to extreme conditions such as large differences between expected and perceived performance. In addition, an integration error tested demonstrated that the model was robust to changes in the time step and integration method used for simulation.

Third, a numerical sensitivity analysis was conducted to evaluate whether model outputs change in important ways when there are errors in parameter estimates. The analysis demonstrated that the model is not overly sensitive to errors in parameter values, but did indicate that accurately estimating certain relationships, such as the impact of interventions on system performance, is crucial to model accuracy. Through human subject experiments, sufficient data can be gathered to estimate these relationships, enabling accurate model replications and predictions.

Finally, through Monte Carlo simulations, the CHAS model was able to characterize the impact of human variability on system performance. The CHAS model is able to provide a system designer with a prediction of not only the average value of system performance or workload, but also the plausible range of performance or workload that could occur. This is beneficial to a system designer who wants to evaluate the impact of system changes on the boundary conditions of system design.

7.1.3 Model Generalizability and Limitations

One of the aims of this research was to create a model that was generalizable to a variety of scenarios where a human operator is collaborating with an automated scheduler for real-time scheduling of multiple vehicles in a dynamic, uncertain environment. These scenarios could include searching a designated area, finding and tracking moving targets, visiting designated locations for delivery or pickup, etc. Regardless of whether an operator is collaborating with an AS for real-time multi-UV control, Air Traffic Control (ATC) planning, rail operations scheduling, manufacturing plant operations, or space satellite control, appropriate trust and better feedback about the automation should lead to superior performance. Many of these domains require scheduling under significant uncertainty, which means that human judgment and adaptability should be leveraged in order to guide automation in a collaborative scheduling process. While the method by which humans add value depends on the specific system being modeled, it was shown in Chapter 6 that the CHAS model can be applied to a variety of system design problems across a number of domains that involve real-time human-automation collaborative scheduling.

In addition, a number of limitations of the CHAS model have been identified through the multi-stage validation process. These limitations were categorized in Chapter 6 into three categories. First, the CHAS model is a computational model of human behavior and decision-making and thus makes a number of simplifications and assumptions about the complexities of human perception, cognition, emotions, and decision-making. However, capturing the interactions and relationships between measureable human behavior and the underlying characteristics of human operators is an ongoing and important endeavor, which always has room for improvement. Second, the CHAS model requires sufficient data to validate and tune the causal relationships throughout the model. Thus, the model is limited to assisting designers who aim to make evolutionary changes to existing systems. Third, the CHAS model in its current form is limited to simulating single operator, moderate to high task load missions, within the definition of real-time human-automation collaborative scheduling presented in Section 2.1. This definition includes a *goal-based* architecture (Clare & Cummings, 2011), where the vehicles are semi-autonomous and with the guidance of the AS, can conduct much of the mission on their own. The human operator only guides the high-level goals of the vehicles, as opposed to guiding each

individual vehicle. In addition, the CHAS model assumes that the operator is physically removed from UV environment, thus there is no direct human-robot interaction. Finally, the CHAS model does not consider vehicle failures, communications delays, and safety policies. These boundary conditions are addressed further in Section 7.3 on future work.

7.2 Model Applications

Four example applications of the CHAS model were presented in Chapter 6, to illustrate how the model could potentially be used by UV system designers. First, designers can use the CHAS model to further investigate the impact of operator trust in the Automated Scheduler (AS) on system performance. It was shown that the optimal level of trust in the AS and the optimal intervention rate to maximize system performance could be estimated, given the tradeoff between operator interventions to coach the suboptimal automation and the impact of operator cognitive overload.

Second, the CHAS model can be used to explore a wider system design space that includes both traditional system components as well as human characteristics, such as the cognitive overload onset point. The model was able to demonstrate that operators with higher tolerances for workload could be prompted to intervene more frequently to enhance system performance without inducing cognitive overload.

Third, the CHAS model can support requirements generation for meeting system design specifications, such as maximum workload levels. System designers could utilize the CHAS model with Monte Carlo simulations to evaluate the impact of different workload specifications or design interventions on both average performance and the expected range of performance. Maximizing the minimum expected performance is a common technique in robust optimization (Bertsimas & Thiele, 2006), and it was shown that this technique could potentially be adopted by designers of collaborative human-automation systems.

Finally, designers can evaluate the impact of different automation characteristics on human behavior and system performance. For example, the concept of "neglect benevolence" (Walker, et al., 2012), where certain decentralized algorithms may need time to reach consensus, was implemented in the CHAS model. The CHAS model replicated the fact that operators who

intervene too frequently when collaborating with this type of automation may negatively impact the performance of the automation.

7.3 Future Work

Three main areas of future work have been identified: model refinements, model extensions, and recommendations for human-automation collaborative scheduling system designers and researchers.

First, there are a number of ways that the existing CHAS model could be refined to provide more accurate simulations of a variety of different systems. These include:

- Modeling additional human characteristics endogenously. While the CHAS model assumed that a number of human time constants (i.e. the time constant for adjusting trust) were static and could be modeled as exogenous parameters, future work should analyze whether these assumptions were valid or whether these characteristics should be modeled endogenously. Also, additional human characteristics such as experience using the system could be explicitly added to the model.
- Adding additional causal relationships to the model. For example, qualitative data from the new human subject experiment indicated that under high workload situations, operators tend to rely on automation in order to reduce their workload. Whether or not this actually indicates an increase in trust is debatable, as it may simply be an increase in reliance due to cognitive overload. While the CHAS model captures the direct impact of high workload on the value that humans can add to system performance, this interesting interaction between high workload and reliance on automation is not currently captured in the model, but should be explored in future work. Additionally, a more sophisticated model of operator decisions to replan would increase model accuracy. This model should also include the fact that creating new search tasks would not have a positive impact system performance if the operator does not replan to assign these search tasks to the team of UVs. Finally, automation learning and adaptation could be modeled in addition to human learning.
- Implementing a non-linear representation of Nonscheduling Task Load (NST). This could replicate the rise and fall of NST that was seen in experimental data. Alternatively, the

CHAS model could be driven with external data that contains accurate NST information for each mission.

- Refining the implementation of the cognitive overload loop. As part of testing the applicability of the CHAS model to systems with different automation characteristics, it was observed that the model does not capture that fact that even ineffective interventions under conditions of cognitive overload can continue to negatively impact decentralized algorithms. Future work should address this limitation of the model.

- Adding additional constructs to the model to capture the initial transients in operator perceptions, behavior, and workload prior to achieving SA. Similarly, the model could be expanded to better capture the differences between novices and experts when performing real-time human-automation collaborative scheduling. Further model structure could also be added to calculate different performance metrics, such as targets found, which are important to real-time human-automation collaborative scheduling missions.

Second, there are a number of ways that the existing CHAS model could be extended to simulate different types of human-automation collaborative scheduling situations. These include:

- Consideration of teams of operators controlling teams of UVs. This would include modeling team communication, the impact of team structure on interaction with the automation, as well as interpersonal trust in addition to simply trust in the automation.

- Modeling the impact of low task load, vigilance missions. All of the data used to build and test the CHAS model came from medium to high task load missions. The CHAS model should be tested on low task load experimental data and refined as necessary to take into account the negative impact of vigilance on human performance.

- The CHAS model could potentially be expanded to include many other features, such as the impact of vehicle failures, communications delays, and safety policies. All of these attributes were excluded from the model boundary to keep the initial model simple. Also, all model validation data came from simulation-based experiments that did not incorporate vehicle failures, communications delays, or safety policies. However, the implementation of multi-UV systems in real-world settings will require an intensive investigation of how both operators and automation respond to all of these attributes.

234

Third, from the results of the new human subject experiment, design recommendations and insights for the creators of scheduling algorithms and the designers of collaborative interfaces were developed:

- A number of the qualitative comments from the human subject experiment (Section 5.7.4) indicated that operators lost trust in the automation because they did not always understand the decisions that the automation was making. Thus, rather than training operators to understand the way the automation makes decisions, automation designers should develop methods that enable the automation to provide reasons for scheduling decisions. Also, interface designers should find innovative methods to display this information to operators.

- The CHAS model can assist researchers in quantitatively assessing the impact of different automated scheduling algorithms. First, the model explicitly represents the contribution of the automation to the performance of the system through the Automation Generated Search Speed variable. This provides the system designer with the ability to quantitatively evaluate the impact of improved automation, such as a faster search algorithm, on the performance of the system, as well as potentially on the operator's trust level, intervention rate, and workload. Second, the model takes as an exogenous parameter the average Length of Time to Replan. If an algorithm designer reduces the time that the AS takes to generate a new schedule, the impact on operator workload and overall system performance can be evaluated. Additionally, it might be shown quantitatively that additional improvements in schedule generation speed beyond a certain point have diminishing returns for operator workload and system performance, and thus some less complex, more predictable, yet potentially "slower" algorithms may, in fact, be fast enough for real-time human-automation collaborative scheduling. Such an approach could indicate whether more effort should be put into reducing computation time versus improving the robustness of the algorithm. Third, the model captures the effect of operator interventions on the human-automation collaboration and thus on system performance. This effect is implemented in the model using an empirically derived, non-linear relationship between the search task rate and human value added which is specific to the AS used in the testbed. As demonstrated in Section 6.1.4, the CHAS model could potentially allow a system designer to investigate the impact of a different AS on human and system performance. The empirical relationship between operator interventions

and the value that the human adds to system performance could be adjusted for a new AS using, for example, data on the neglect time necessary for algorithm stabilization (Walker, et al., 2012).

- Both quantitative and qualitative data from the experiment indicated that the reference line on the performance plot may have been set too high for the test subjects who experienced High Real Time Priming, leading to a sense that the goal was unachievable and lower ratings of confidence. It also led to a shifting of responsibility, where some operators placed all blame for poor system performance on the automation. In contrast, another group of test subjects had a low reference line on the performance plot which provided positive reinforcement to operators, increasing their confidence. Future work should further explore the impact of expectation setting, including whether a more achievable reference performance curve could cause the intended effect of pushing operators to intervene more frequently without lowering operator confidence.

- Future work should further explore the impact of gaming on the behavior and performance of UV operators. A more detailed and structured demographic questionnaire on gaming activity could enable deeper analysis of the impact of different types of gaming, i.e. shooter vs. real-time strategy games.

- It is crucial to accurately capture the impact of delays in human perception of system performance when modeling human-automation collaborative scheduling of multiple UVs. Thus, future research should further explore the impact of real-world communications delays, such as delaying or restricting information displayed to the operator, on operator behavior, trust, and system performance. Also, future research could utilize eye tracking devices for a more precise estimate of human perception patterns and delays.

7.4 Contributions

The objective of this thesis was to develop and validate a computational model of real-time human-automation collaborative scheduling of multiple UVs that could be used to predict the impact of changes in system design and operator training on human and system performance. In striving to achieve this objective, several contributions have been made to the domain of human-automation collaboration research. These include theoretical, modeling, and experimental design contributions, which are discussed below. Additionally, this thesis begins to fill an

236

interdisciplinary research gap among the fields of human factors, autonomy, and complex system modeling by adapting System Dynamics modeling techniques to model human-automation collaboration for the control of multiple semi-autonomous UVs.

Three research questions were posed in Chapter 1 to address this objective. By answering the first question, "What are the major attributes and interactions that a model of real-time human-automation collaborative scheduling must capture?" this thesis resulted in the:

- Identification of six attributes that are important to consider when modeling real-time human-automation collaborative scheduling, providing a theoretical basis for the model developed in this thesis. The attributes are: attention allocation and situation awareness, cognitive workload, trust in automation, human learning, automation characteristics, and human value-added through interventions.

- Development and initial validation of a dynamic hypothesis of human-automation collaborative scheduling: if operators can either a) anchor to the appropriate trust in the automation and expectations of performance earlier in the mission and/or b) adjust their trust and expectations faster through better feedback about the automation, then system performance should improve.

By answering the second question, "Can the model be used to predict the impact of changes in system design and operator training on human and system performance?" this thesis resulted in the:

- Implementation and initial validation of a System Dynamics (SD) simulation model, the Collaborative Human-Automation Scheduling (CHAS) model. The CHAS model utilizes SD constructs (stocks, flows, causal loops, time delays, feedback interactions) to model real-time human-automation collaborative scheduling of multiple UVs.

- Successful quantitative prediction of the impact of *a priori* priming of trust in automation on test subjects who play computer and video games frequently. These gamers were found to have a higher propensity to over-trust automation and lowering their initial trust level helped them recognize the imperfections in the automation. This encouraged them to intervene more frequently to coach the suboptimal automation, leading to an improvement in system performance.

Finally by answering the third set of questions, "What level of accuracy can be expected of this model? How does it compare to other relevant models? What are the boundary conditions of the model?" this thesis resulted in the:

- Quantification of the non-linear relationships, in a very specific instantiation, for human trust in automation, the rate of human interventions to guide automation, and the impact of intervention rate on automation and system performance. These relationships were quantified using data from two different simulation testbeds, demonstrating the positive effects of initial increases in intervention rate along with the diminishing returns of high rates of intervention.
- Novel use of a real-time survey to gather data on operator perceptions of performance, expectations of how well the system should be performing, and trust in automation. This survey was able to gather valuable data to evaluate model assumptions without substantially increasing operator workload.

Real-time scheduling in uncertain environments is crucial to a number of domains, especially UV operations. With the ever-increasing demand for UVs for both military and commercial purposes, inverting the operator-to-vehicle ratio will become necessary. Real-time scheduling for multiple UVs in uncertain environments will require the computational ability of optimization algorithms combined with the judgment and adaptability of human supervisors. Despite the potential advantages of human-automation collaboration, inappropriate levels of operator trust, high operator workload, and a lack of goal alignment between the operator and automation can cause lower system performance and costly or deadly errors. The CHAS model can support designers of future UV systems working to address these challenges by simulating the impact of changes in system design and operator training on human and system performance. This could help designers save time and money in the design process, enable the exploration of a wider trade space of system changes than is possible through prototyping or experimentation, and assist in the real-world implementation of multi-vehicle unmanned systems.

Appendix A: Model Reduction Process

The CHAS model underwent significant iteration, including a model reduction process. The original CHAS model (Figure 77) had many more parameters, modules, and feedback loops than the parsimonious model presented in Chapter 3. The goal of this reduction process was to determine which modules were necessary through progressive "loop knockout" tests. Known as "behavior anomaly" testing (Sterman, 2000), if a loop knockout test generates anomalous behavior, it suggests the importance of the loop. The reduction process consisted simply of evaluating whether there was an increase, decrease, or no change in the goodness of fit for seven variables that could be compared to experimental data, using the data set described in Chapter 3.2. These variables were: area coverage, length of time to replan, utilization, the probability of performing a what-if assignment, the probability of modifying the objective function of the AS, search task rate, and replan rate. Three major modules that were removed following loop knockout tests are presented below.

Short-Term Learning

The first module to be removed was the "Short-Term Learning" module. The original intention of this module was to capture learning effects as operators became more comfortable using the interface and evaluating schedules. Both the experimental data set presented in Chapter 3.2 and the experiment presented in Chapter 5 had data showing that operators became faster at replanning over time through the mission. Thus, the Short-Term Learning module was important to the fit of the model with regards to length of time to replan. Once the module was removed, however, there was no change in the fit of the model to the area coverage performance metric or probability of a what-if assignment. The fit of the utilization and probability of modifying the objective function fit decreased, while the search task rate and replan rate fit improved. Thus, it was decided to remove the Short-Term Learning module, as removing it did not change the fit to the primary performance metric and actually improved the fit to important intervention metrics.

Figure 77. Original CHAS model with removed and modified modules shown.
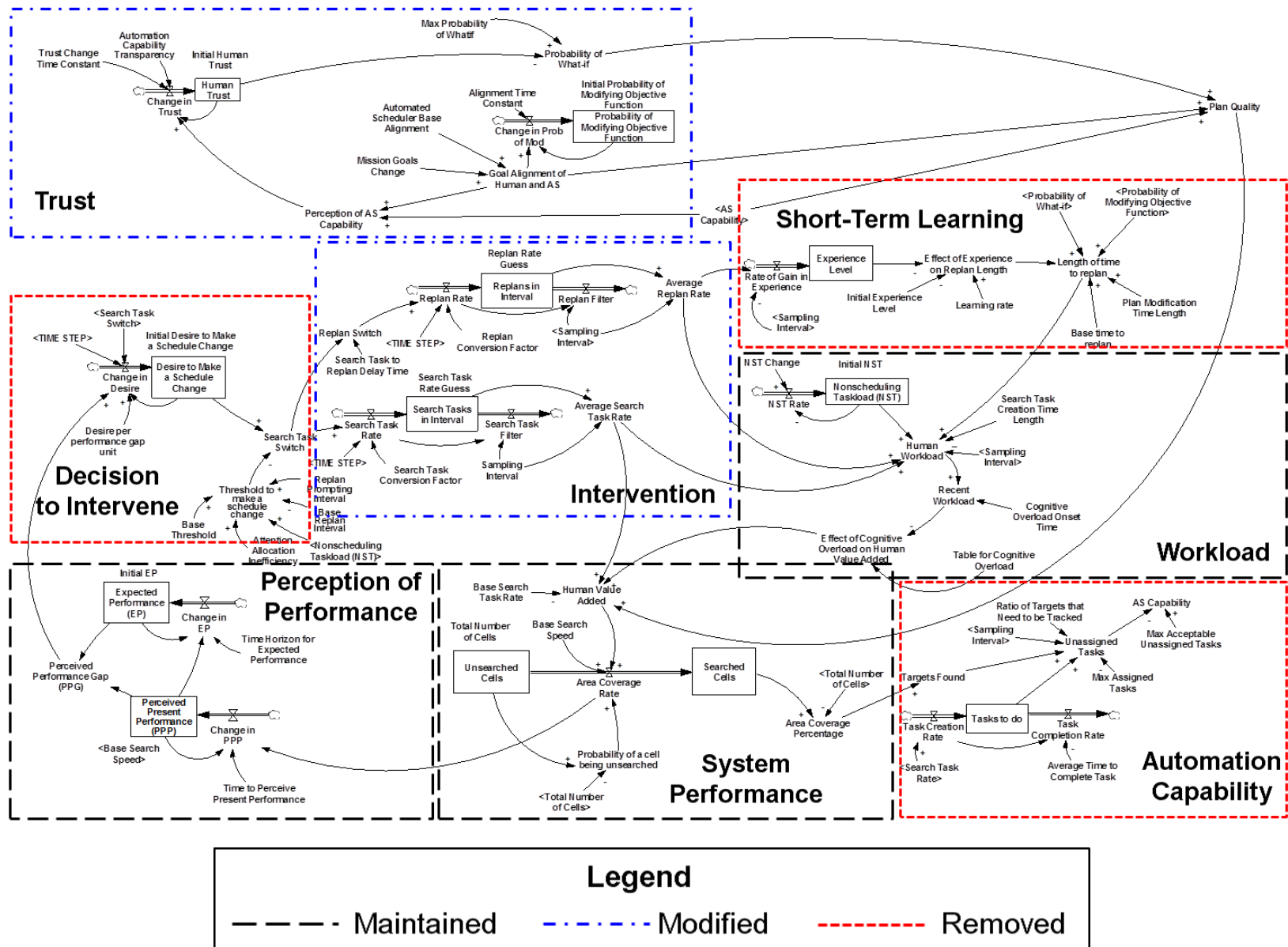
By removing the Short-Term Learning module, it reduced the number of exogenous parameters in the model by 2 and the number of endogenous variables by 3. As opposed to modeling the length of time to replan as an endogenous variable that changed over time based on a learning curve, the CHAS model now captured the length of time to replan as a static exogenous parameter. It should be noted that the Short-Term Learning module could always be re-instated in the model if the system being modeled had a significant short-term learning curve.

Automation Capability

The second module to be removed was the "Automation Capability" module. This module assumed that the operator perceived the capability of the automation separately from the performance of the overall system, which is likely true, but a fine distinction. Written feedback from test subjects (Clare, Cummings, How, et al., 2012) indicated that operators often perceived the capability of the AS based on the tasks that the AS chose to assign in resource constrained situations. For example, one operator wrote, "I did not always understand decisions made by the automated scheduler…namely it would not assign tasks…while some vehicles were seemingly idle." Another operator wrote "The automated scheduler would assign a different UAV [to a task] than I would have picked."

Thus, this module attempted to capture the capability of the AS by relating the number of unassigned tasks to AS capability. The AS often could not assign all available tasks due to UV resource shortages, which is representative of real world constraints. In order to model the number of unassigned tasks, the module first calculated the number of targets that had been found based on an empirical relationship between area coverage and targets found. It also maintained a stock of "Tasks to Do" based on the search task creation rate and the average time to complete search tasks. Combining the number of targets to track with the Tasks to Do stock enabled estimation of the number of unassigned tasks and perceived automation capability.

In the pursuit of a parsimonious model, it was desirable to evaluate the potential impact of removing the Automation Capability module. However, in order to remove the Automation Capability module, the Trust module also needed to be modified. Instead of calculating the perception of AS capability from the number of unassigned tasks, it was modified to use the Perceived Performance Gap (PPG), as described in Chapter 3. The model now assumes that the

operator's perception of the capability of the AS is directly related to the operator's overall perceptions and expectations of system performance, as opposed to a distinct perception of the automation capability based on the number of unassigned tasks in the schedule.

In the test that removed the Automation Capability module and modified the Trust module, there was no change in the fit of the model to the area coverage performance metric, utilization, search task rate, or replan rate. The fit to the probability of modifying the objective function fit and probability of a what-if assignment decreased. However, the cluster analysis presented in Section 3.2 found no significant differences in the probability of modifying the objective function or the probability of a what-if assignment between high and low performers. While these may be important measures of operator interaction with the AS, there was a lack of evidence that these specific interventions influenced the primary performance metric in the experimental data, and thus they were removed from the model. This specific change reduced the number of exogenous parameters in the model by 4 and the number of endogenous variables by 4.

With the probability of modifying the objective function fit and probability of a what-if assignment no longer in the model, it was decided to permanently remove the Automation Capability module and modify the Trust module to simplify the model. By removing the Automation Capability module, it reduced the number of exogenous parameters in the model by 5 and the number of endogenous variables by 7. However, the relationship between PPG and Perceived Automation Capability in the Trust module required two new exogenous parameters to define the non-linear logit relationship (Section 3.4.4).

Decision to Intervene

The final major change to the model involved removing the Decision to Intervene module and simplifying the Intervention module. A key assumption in the original Decision to Intervene module was that the operator's desire to make a schedule change could be modeled as a continuously increasing function based on his or her Perceived Performance Gap (PPG), followed by a discrete decision to intervene based on a threshold. There is some precedence in the SD literature for modeling a decision-making process in this manner, for organizational change (Sastry, 1995) and in high-risk, time-critical environments such as emergency operating rooms (Rudolph, et al., 2009). Thus, the Decision to Intervene module contained novel mixed

continuous/discrete modeling elements. The model represented the operator's Desire to Make a Schedule Change as a stock that increased when there was a positive PPG, when the operator's expectation of performance was higher than the perceived present performance. Once the operator's desire to intervene was greater than a threshold, a discrete intervention event occurred, causing the operator's Desire to reset to zero. The interventions that were chosen for this model were creating a search task and replanning. The model components shown in the original Intervention module convert the discrete intervention events back into continuous variables through the use of a pipeline delay function.

Once again, in the pursuit of a parsimonious model, it was desirable to evaluate the potential impact of removing the Decision to Intervene module. In order to remove the Decision to Intervene module and simplify the Intervention module, two major changes were made. First, the model was modified to have Human Trust drive decisions to intervene in terms of search task creation and replanning. This removed the need for a "Desire to Intervene" stock, which did not have real-world meaning and would be impossible to measure. Second, the model calculated a continuous "search task rate" and "replan rate" directly from Human Trust, without the use of continuous to discrete to continuous conversions.

In the test that removed the Decision to Intervene module and simplified the Intervention module, there was no change in the fit of the model to the area coverage performance metric or to utilization. However, the fit to search task rate and replan rate both improved. Thus, this change was made permanent. This reduced the number of exogenous parameters in the model by 12 and the number of endogenous variables by 11. However, the relationship between Human Trust and Search Task Rate in the Intervention module required four new exogenous parameters to define the non-linear logit relationship (Section 3.4.5).

Other additional minor changes to the model were made, but the major modules and feedback loops remained the same from this point on to the final parsimonious model presented in Chapter 3.

244

Appendix B: Time Series Data Analysis

This section presents additional data and statistics for time series data analysis discussed in Section 3.2.2. Two separate clustering analyses were conducted, with the first analysis using total area coverage by the end of the mission as the clustering metric, and the second analysis using total number of targets found by the end of the mission. A hierarchical clustering was conducted using Ward's Method to determine the number of clusters. Afterwards, the k-means algorithm was used to assign missions to clusters. Following clustering, two performance metrics, one measure of workload, and five operator action measures were compared between the high and low performance clusters for each performance metric. These measures were: area coverage, targets found, utilization, length of time to replan, the probability of performing a what-if assignment, the probability of modifying the objective function of the AS, search task rate, and replan rate.

The first analysis used area coverage by the end of the mission as the clustering metric. Three clusters were identified: low, medium, and high performance groups. Of the total 60 missions, there were 11 missions in the High Performance cluster and 26 missions in the Low Performance cluster. A comparison of the performance metrics between the high and low clusters is shown in Figure 78. Workload and operator action metrics are shown in Figure 79. All 8 metrics were evaluated with a repeated measures ANOVA and the results are summarized in Table 15. There were significant differences between the performance clusters in terms of 3 of the operator action metrics: length of time to replan, search task rate, and replan rate.
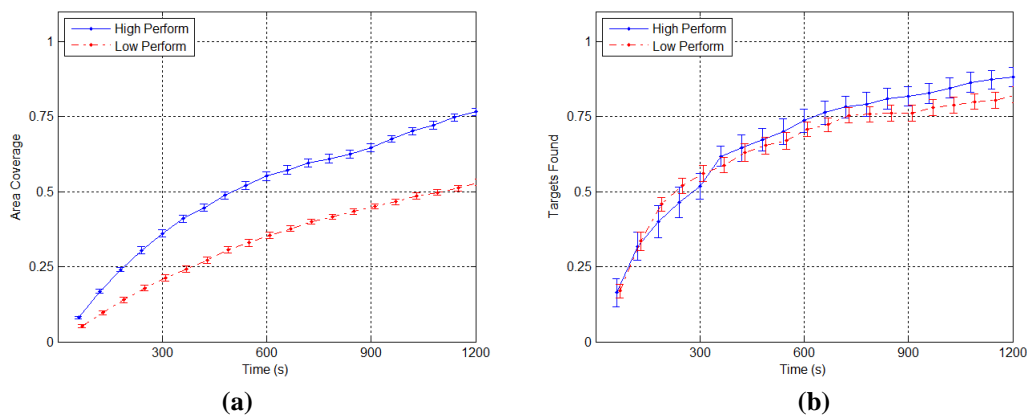
Figure 78. Performance metrics for cluster analysis based on area coverage. (a) Area Coverage. (b) Targets Found. Standard Error bars shown.
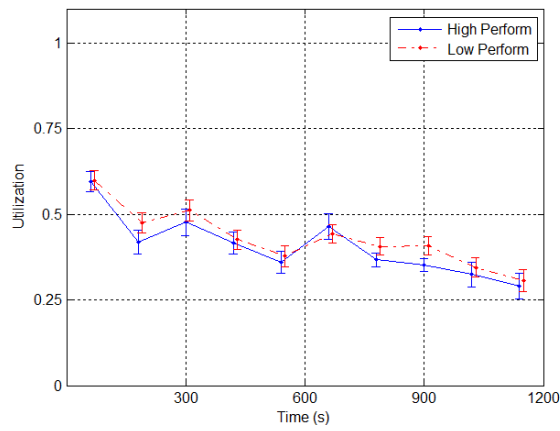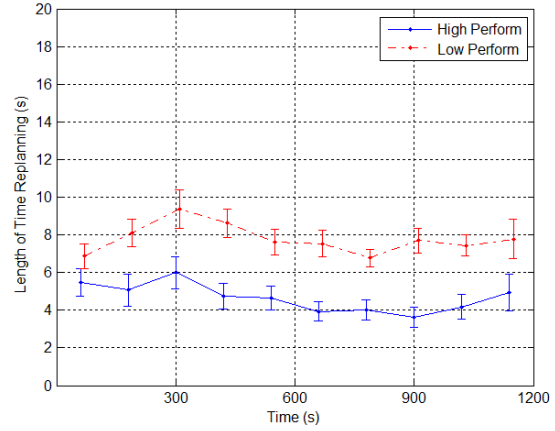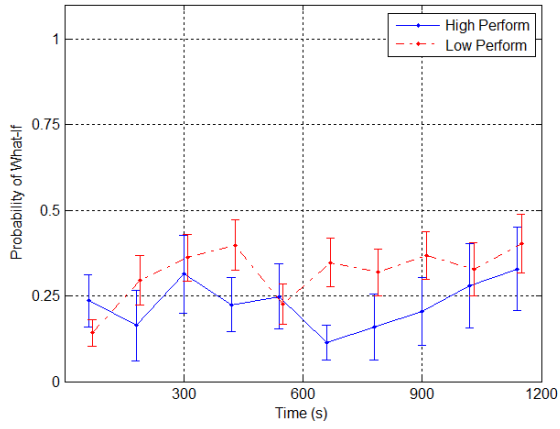
Figure 79. Workload and operator action metrics for cluster analysis based on area coverage. (a) Utilization. (b) Length of time to replan. (c) Probability of performing a what-if assignment. (d) Probability of modifying the objective function of the AS. (e) Replan rate. (f) Search task rate. Standard Error bars shown.

Table 15. Summary of repeated ANOVA results for cluster analysis based on area coverage.

Metric	Main Effect for Performance Cluster (2 levels: High and Low)	Main Effect for Time (10 time intervals)	Interaction Effect Between Time and Performance Cluster
Area Coverage	$F(1,35) = 145.801$, $p < 0.001$	$F(9,315) = 1652.294$, $p < 0.001$	$F(9,315) = 46.441$, $p < 0.001$
Targets Found	$F(1,35) = 0.300$, $p = 0.587$	$F(9,315) = 259.299$, $p < 0.001$	$F(9,315) = 1.744$, $p = 0.078$
Utilization	$F(1,35) = 2.472$, $p < 0.125$	$F(9,315) = 16.215$, $p < 0.001$	$F(9,315) = 0.580$, $p = 0.814$
Length of time to replan	$F(1,27) = 5.910$, $p = 0.022$	$F(9,243) = 0.402$, $p = 0.933$	$F(9,243) = 0.625$, $p = 0.775$
Probability of performing a what-if assignment	$F(1,27) = 3.799$, $p = 0.062$	$F(9,243) = 1.060$, $p = 0.393$	$F(9,243) = 1.020$, $p = 0.425$
Probability of modifying the objective function of the AS	$F(1,27) = 0.096$, $p = 0.759$	$F(9,243) = 2.272$, $p = 0.018$	$F(9,243) = 0.837$, $p = 0.583$
Replan rate	$F(1,35) = 10.485$, $p = 0.003$	$F(9,315) = 2.705$, $p = 0.005$	$F(9,315) = 1.520$, $p = 0.140$
Search task rate	$F(1,35) = 18.697$, $p < 0.001$	$F(9,315) = 2.517$, $p = 0.009$	$F(9,315) = 1.299$, $p = 0.236$

The second analysis used targets found by the end of the mission as the clustering metric. Three clusters were identified: low, medium, and high performance groups. Of the total 60 missions, there were 14 missions in the High Performance cluster and 13 missions in the Low Performance cluster. A comparison of the performance metrics between the high and low clusters is shown in Figure 80. Workload and operator action metrics are shown in Figure 81. All 8 metrics were evaluated with a repeated measures ANOVA and the results are summarized in Table 16. There were no significant differences found between the performance clusters in terms of the 5 operator action metrics.


(a)   **(b)**

Figure 80. Performance metrics for cluster analysis based on targets found. (a) Area Coverage. (b) Targets Found. Standard Error bars shown.
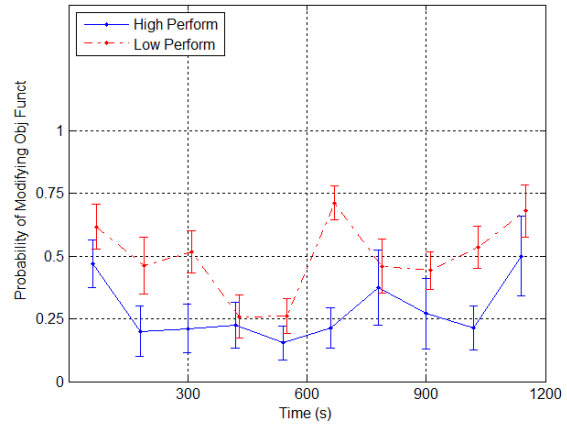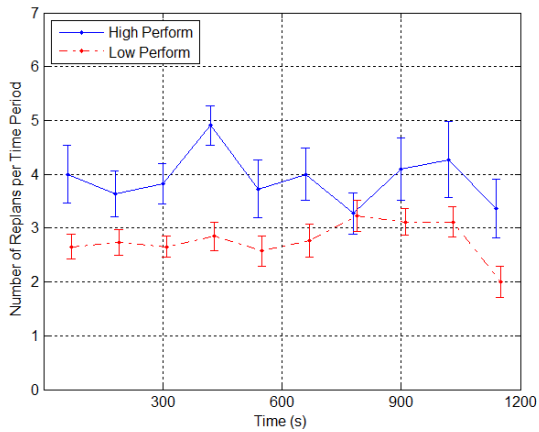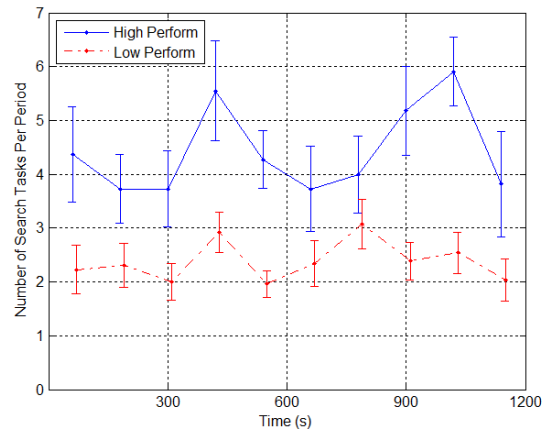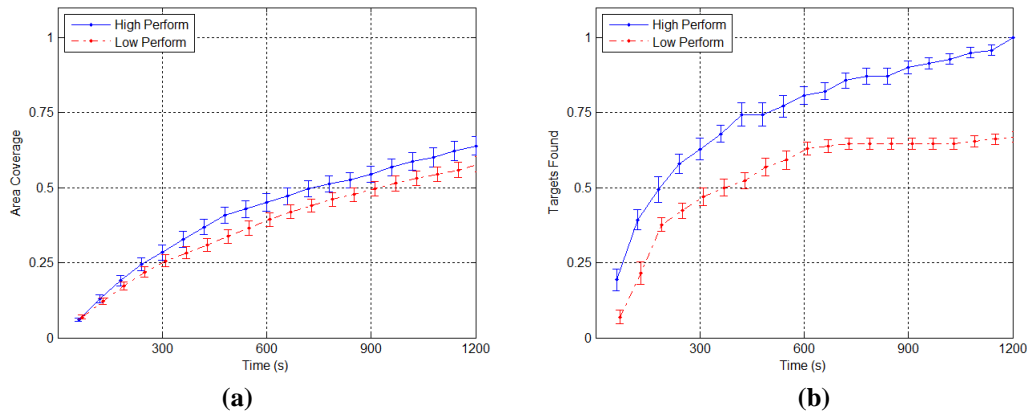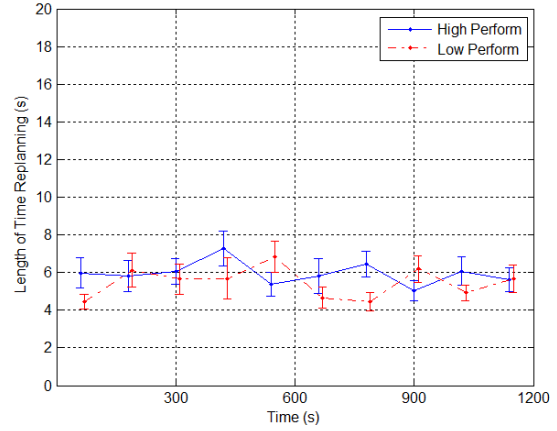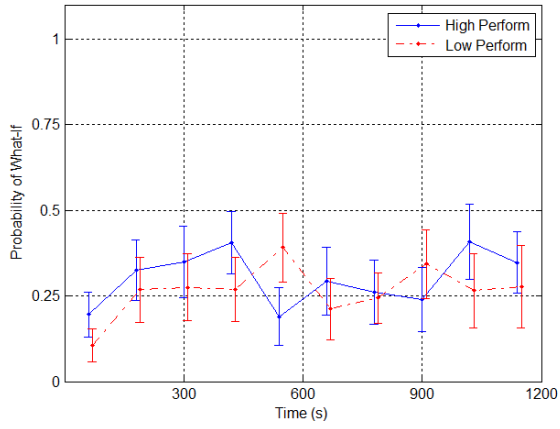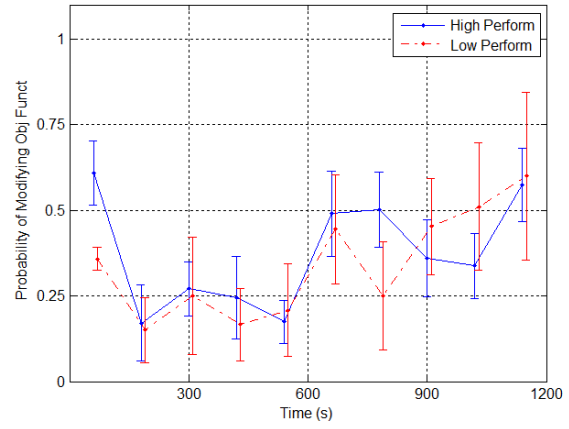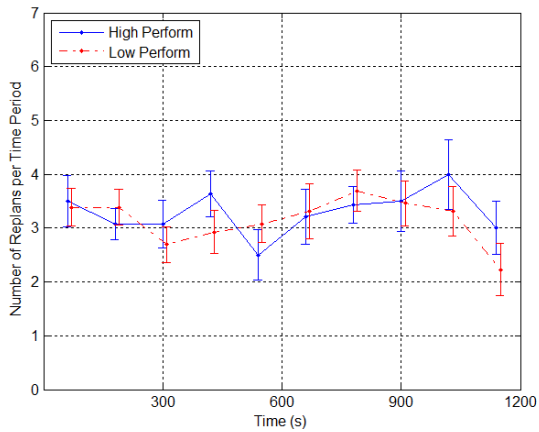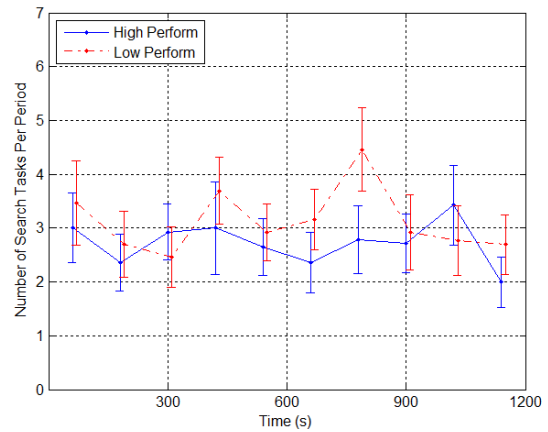
Figure 81. Workload and operator action metrics for cluster analysis based on targets found. (a) Utilization. (b) Length of time to replan. (c) Probability of performing a what-if assignment. (d) Probability of modifying the objective function of the AS. (e) Replan rate. (f) Search task rate. Standard Error bars shown.
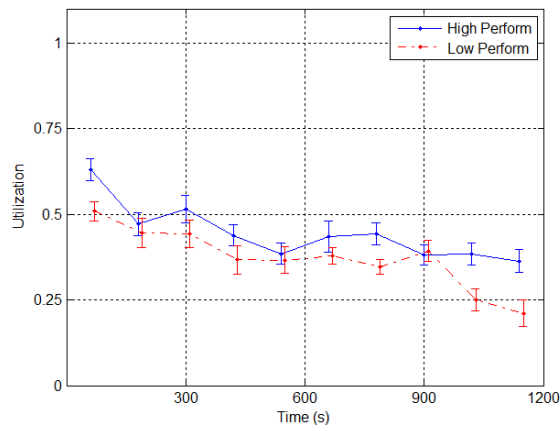
Table 16. Summary of repeated ANOVA results for cluster analysis based on targets found.

Metric	Main Effect for Performance Cluster (2 levels: High and Low)	Main Effect for Time (10 time intervals)	Interaction Effect Between Time and Performance Cluster
Area Coverage	$F(1,25) = 2.032$, $p = 0.166$	$F(9,225) = 668.514$, $p < 0.001$	$F(9,225) = 3.047$, $p = 0.002$
Targets Found	$F(1,25) = 62.561$, $p < 0.001$	$F(9,225) = 281.269$, $p < 0.001$	$F(9,225) = 5.327$, $p < 0.001$
Utilization	$F(1,25) = 5.609$, $p = 0.026$	$F(9,225) = 16.721$, $p < 0.001$	$F(9,225) = 1.811$, $p = 0.067$
Length of time to replan	$F(1,20) = 0.388$, $p = 0.541$	$F(9,180) = 0.942$, $p = 0.490$	$F(9,180) = 1.248$, $p = 0.268$
Probability of performing a what-if assignment	$F(1,20) = 0.072$, $p = 0.791$	$F(9,180) = 1.023$, $p = 0.423$	$F(9,180) = 1.206$, $p = 0.294$
Probability of modifying the objective function of the AS	$F(1,20) = 2.855$, $p = 0.107$	$F(9,180) = 3.027$, $p = 0.002$	$F(9,180) = 1.234$, $p = 0.277$
Replan rate	$F(1,25) = 0.092$, $p = 0.764$	$F(9,225) = 2.692$, $p = 0.005$	$F(9,225) = 1.257$, $p = 0.262$
Search task rate	$F(1,25) = 0.460$, $p = 0.504$	$F(9,225) = 1.290$, $p = 0.243$	$F(9,225) = 0.898$, $p = 0.528$

Appendix C: CHAS Model Equations and Parameters

System Performance Module

Human Value Added=
        Effect of Cognitive Overload on Human Value Added*Effect of Search Tasks on Human Value Added
Units: Cells/Second

Area Coverage Rate=
        Probability of a cell being unsearched*(Automation Generated Search Speed+Human Value Added)
Units: Cells/Second

Area Coverage Percentage=
        Searched Cells/Total Number of Cells
Units: % Area Covered

Probability of a cell being unsearched=
        Unsearched Cells/Total Number of Cells
Units: %

Automation Generated Search Speed=
        2.9
Units: Cells/Second

Total Number of Cells=
        4150
Units: Cells

Unsearched Cells=
        INTEG (-Area Coverage Rate, Total Number of Cells)
Units: Cells

Searched Cells=
        INTEG (Area Coverage Rate, 1)
Units: Cells

Perception of Performance Module

"Perceived Present Performance (PPP)"=
        SMOOTH3(Area Coverage Rate , Time to Perceive Present Performance)
Units: Cells/Second

Time to Perceive Present Performance=
        150
Units: Seconds

Initial EP=
        9.8
Units: Cells/Second

"Expected Performance (EP)"=
        INTEG (Change in EP, Initial EP)
Units: Cells/Second

Change in EP=
　　　("Perceived Present Performance (PPP)" - "Expected Performance (EP)")/Time Horizon for Expected
　　　Performance
Units: Cells/(Second*Second)

Time Horizon for Expected Performance=
　　　190
Units: Seconds

"Perceived Performance Gap (PPG)"=
　　　("Expected Performance (EP)"-"Perceived Present Performance (PPP)")/"Expected Performance (EP)"
Units: % PPG

Trust Module

Perceived Automation Capability=
　　　(1-1/(1+EXP(-4*PPG Max Slope*("Perceived Performance Gap (PPG)"-Base PPG))))
Units: % Automation Capability

Base PPG=
　　　13
Units: % PPG

PPG Max Slope=
　　　3
Units: Dimensionless

Change in Trust=
　　　(Perceived Automation Capability-Human Trust)/Trust Change Time Constant
Units: % Trust/Second

Human Trust=
　　　INTEG (Change in Trust, Initial Human Trust)
Units: % Trust
Min: 0%
Max: 100%

Initial Human Trust=
　　　88
Units: % Trust

Trust Change Time Constant=
　　　410
Units: Seconds

Intervention Module

Search Task Rate=
　　　(Max Search Task Rate-Min Search Task Rate)*(1-1/(1+EXP(-4*Trust Max Slope*(Human Trust\
　　　　　-Midpoint Trust))))+Min Search Task Rate
Units: Search Tasks Per Two Minute Interval

252

Max Search Task Rate=
        8
Units: Search Tasks Per Two Minute Interval

Min Search Task Rate=
        1.9
Units: Search Tasks Per Two Minute Interval

Midpoint Trust=
        25
Units: % Trust

Trust Max Slope=
        2
Units: Search Tasks per Two Minute Interval/% Trust

Replan Rate=
        Search Task Rate*Number of Replans per Search Task
Units: Replans per Two Minute Interval

Number of Replans per Search Task=
        1.1
Units: Replans/Search Tasks

Effect of Search Tasks on Human Value Added=
        (Max Human Value Added-Min Human Value Added)/(1+EXP(-4*Search Task Rate Max Slope*(\
                Search Task Rate-Base Search Task Rate)))+Min Human Value Added
Units: Cells/Second

Max Human Value Added=
        2
Units: Cells/Second

Min Human Value Added=
        -1
Units: Cells/Second

Base Search Task Rate=
        2.85
Units: Search Tasks Per Two Minute Interval

Search Task Rate Max Slope=
        0.6
Units: Cells/(Second*Search Tasks per Two Minute Interval)

Workload Module

Human Workload=
        MIN(1,((Search Task Rate*Search Task Creation Time Length+Replan Rate*Length of Time to Replan\
)/Sampling Interval)+"Nonscheduling Task load (NST)")
Units: % Utilization

Search Task Creation Time Length=
        2.8
Units: Seconds/Search Task

Length of Time to Replan=
        6.5
Units: Seconds/Replan

"Nonscheduling Task load (NST)"=
        MAX(0,MIN(1,Initial NST+NST Rate*Time))
Units: % Utilization

NST Rate=
        -0.0002
Units: % Utilizaiton/Second

Initial NST=
        30
Units: % Utilizaiton

Sampling Interval=
        120
Units: Seconds

Effect of Cognitive Overload on Human Value Added=
        Table for Cognitive Overload(Human Workload)
(see table function below)
Units: Dimensionless

Table for Cognitive Overload(
        (0,1.05),(0.7,1),(0.75,0.9),(0.8,0.7),(0.85,0.4),(0.9,0.1),(0.95,0.05),(1,0))
Units: Dimensionless

Simulation Control Parameters

FINAL TIME = 1200
Units: Seconds

INITIAL TIME = 0
Units: Seconds

TIME STEP = 0.125
Units: Seconds

254

Appendix D: Nonscheduling Task load Analysis

This section presents additional figures and statistics for the analysis of Nonscheduling Task load (NST) discussed in Section 3.4.6. There were three activities in the OPS-USERS testbed that were considered to be nonscheduling activities. First, the operator had to visually identify or re-designate targets. This was done through the pop-up window shown in Figure 82a. Second, the operator had to approve weapons launches on hostile targets, through the pop-up window shown in Figure 82b. These windows would appear over the Map Display (Figure 4) when the operator was prompted to conduct these visual identification tasks. Third, the operator was responsible for reading and answering chat messages, through the chat window shown in Figure 83. When a new chat message arrived, the window blinked and an auditory alert played.
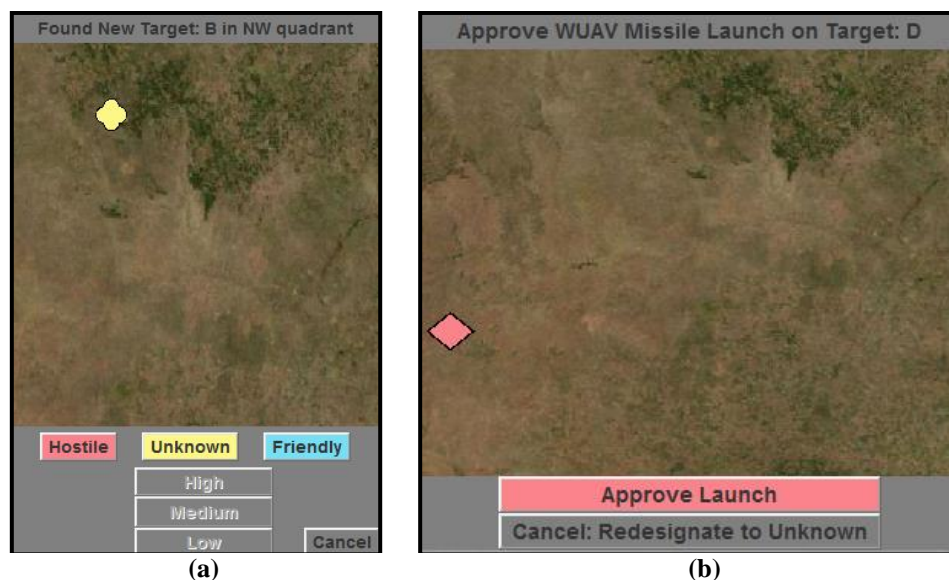

(a)   (b)

Figure 82. Pop-up windows for (a) target identification/re-designation and (b) approving weapons launch.
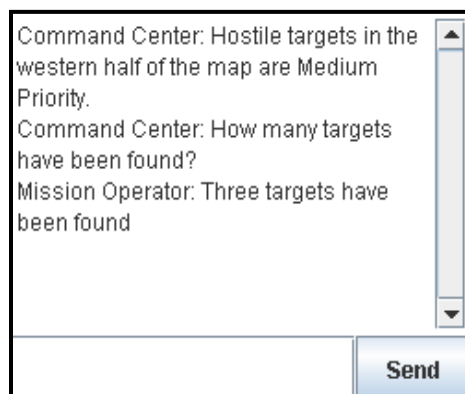

Figure 83. OPS-USERS chat window.

255

The required utilization due to NST was calculated from aggregate experimental data (Section 3.2.2), using two-minute intervals over the twenty minute experiment. Descriptive statistics for the required utilization due to NST in each interval are shown in Table 17.

Table 17. Descriptive statistics for required utilization due to Nonscheduling Task load (NST).

Interval (min)	0-2	2-4	4-6	6-8	8-10	10-12	12-14	14-16	16-18	18-20
Mean	38.9%	24.8%	26.7%	16.1%	16.4%	22.6%	17.1%	17.4%	9.5%	10.3%
Median	35.7%	23.2%	24.6%	12.7%	15.3%	20.1%	14.9%	16.5%	8.6%	9.3%
Min	20.0%	5.9%	8.3%	4.2%	4.2%	4.2%	4.3%	4.2%	0.0%	0.0%
Max	73.6%	46.1%	66.4%	61.7%	49.0%	59.4%	40.3%	45.7%	30.5%	28.4%
St. Dev	12.7%	9.0%	11.7%	10.8%	8.7%	11.8%	8.0%	8.6%	6.6%	7.3%
SE	1.6%	1.2%	1.5%	1.4%	1.1%	1.5%	1.0%	1.1%	0.9%	0.9%

A repeated measures ANOVA of the required utilization due to NST was conducted. There were no between-subject variables and there were 10 levels of time (representing each interval). The repeated measures ANOVA of the required utilization due to NST indicated a significant effect for time, $F(9,531) = 55.140$, $p < 0.001$.

Finally, the same analysis was conducted for data sets from four different OPS-USERS experiments, with aggregate results for all operators shown in Figure 84. The first experiment had medium workload, 10-minute long missions (Cummings, Clare, et al., 2010), with NST data shown in Figure 84a. Second, the analysis was conducted for a high task load experiment, with 10-minute long missions, but double the number of targets to identify, more chat messages, and faster UVs (Section 4.2.2), with NST data shown in Figure 84b. The third experiment had 20-minute long, medium workload missions, where operators could adjust the objective function of the AS (Section 3.2), with NST data shown in Figure 84c. Finally, the NST analysis was conducted for the experiment presented in Chapter 5 of this thesis to validate the CHAS model, as shown in Figure 84d. The required utilization due to NST is shown in red and the self-imposed utilization from scheduling activities (creating search tasks and replanning) is shown in blue stripes.

In Figure 84a, c, and d, there appears to be a roughly linear decline in utilization due to NST over time. It is notable, however, how different the required utilization due to NST is among the various experiments, all of which use the same OPS-USERS testbed, but have different

256

experimental conditions, numbers of targets, speeds of UVs, chat message questions, and Rules of Engagement. Having an accurate representation of NST is thus important to the CHAS model's accuracy for calculating operator workload.
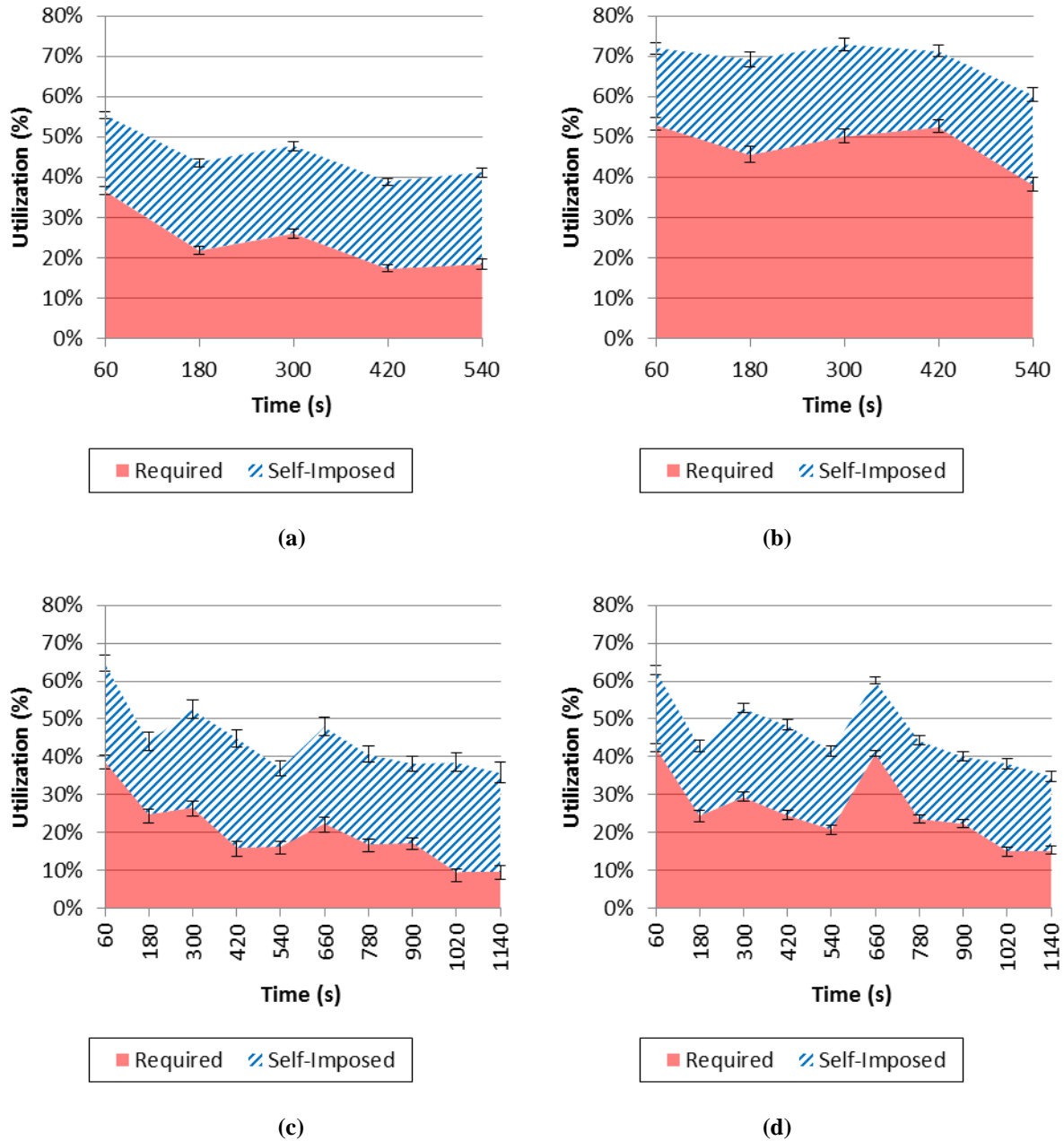
Figure 84. Utilization due to self-imposed scheduling activities and required utilization due to Nonscheduling Task load (NST) for different OPS-USERS experiments. (a) Medium workload replan prompting experiment. (b) High task load experiment. (c) Dynamic objective function experiment. (d) CHAS validation experiment. Standard error bars are shown.

Appendix E: Model Validation Test Results

Sections 4.1.3 and 4.1.4 described two types of model structure tests: extreme conditions tests and integration error tests. Results for these tests are presented below.

Extreme Conditions Tests

Extreme conditions tests "ask whether models behave appropriately when the inputs take on extreme values" (Sterman, 2000, p. 869). Three extreme conditions tests were conducted, with a focus on: area coverage rate, Perceived Performance Gap (PPG), and human workload. In each test, the behavior of the model under extreme inputs was evaluated for the variable of interest, in addition to an evaluation of the resulting operator interventions and system performance.

First, Section 4.1.3 presented the behavior of the model when the Automation Generated Search Speed parameter was set to extremely low and high values. It was shown that the overall system performance calculated by the model was appropriate even under these extreme conditions (Figure 26).

For the second test, operator expectations of performance were initialized to an extremely high level. Initial EP was set to 10x the baseline value in the simulation, causing the Perceived Performance Gap (PPG) to go to extremely high levels (100% PPG) compared to the baseline simulation (Figure 85a). However, the Perceived Automation Capability (Figure 85b) went no lower than 0% as designed. Through the use of non-linear logit curves to define the relationships between variables such as PPG and Perceived Automation Capability (Section 3.4.4), the model is robust to extreme inputs.

Additionally, operator expectations of performance were initialized to an extremely low level. Initial EP was set to 10% of the baseline value in the simulation, causing the Perceived Performance Gap (PPG) to go to extremely negative levels compared to the baseline simulation (Figure 85a). This indicates that the simulated operator perceived that the system was performing far better than expected. However, the Perceived Automation Capability (Figure 85a) went no higher than 100% as designed, once again due to the non-linear logit curve defining te relationship between PPG and Perceived Automation Capability.
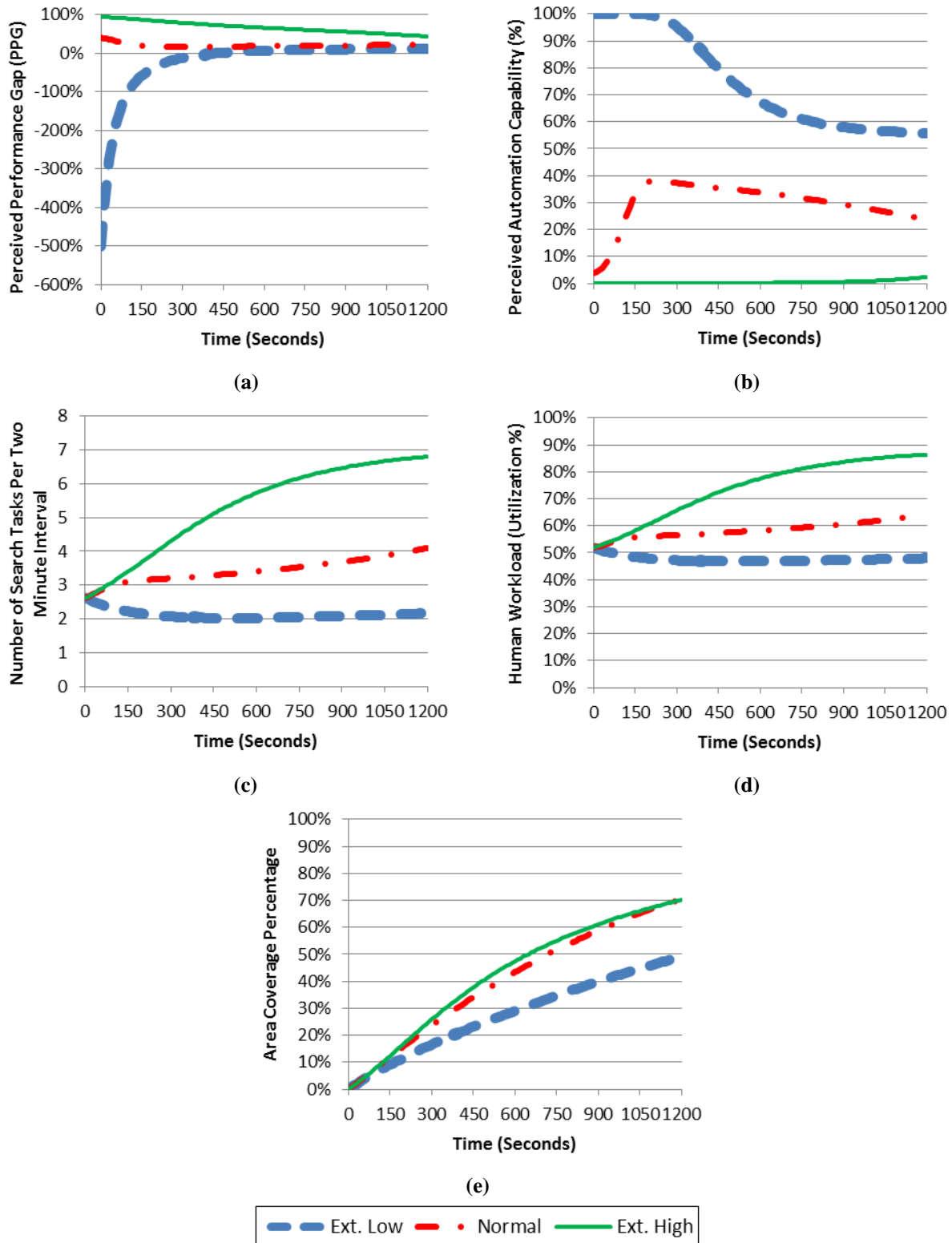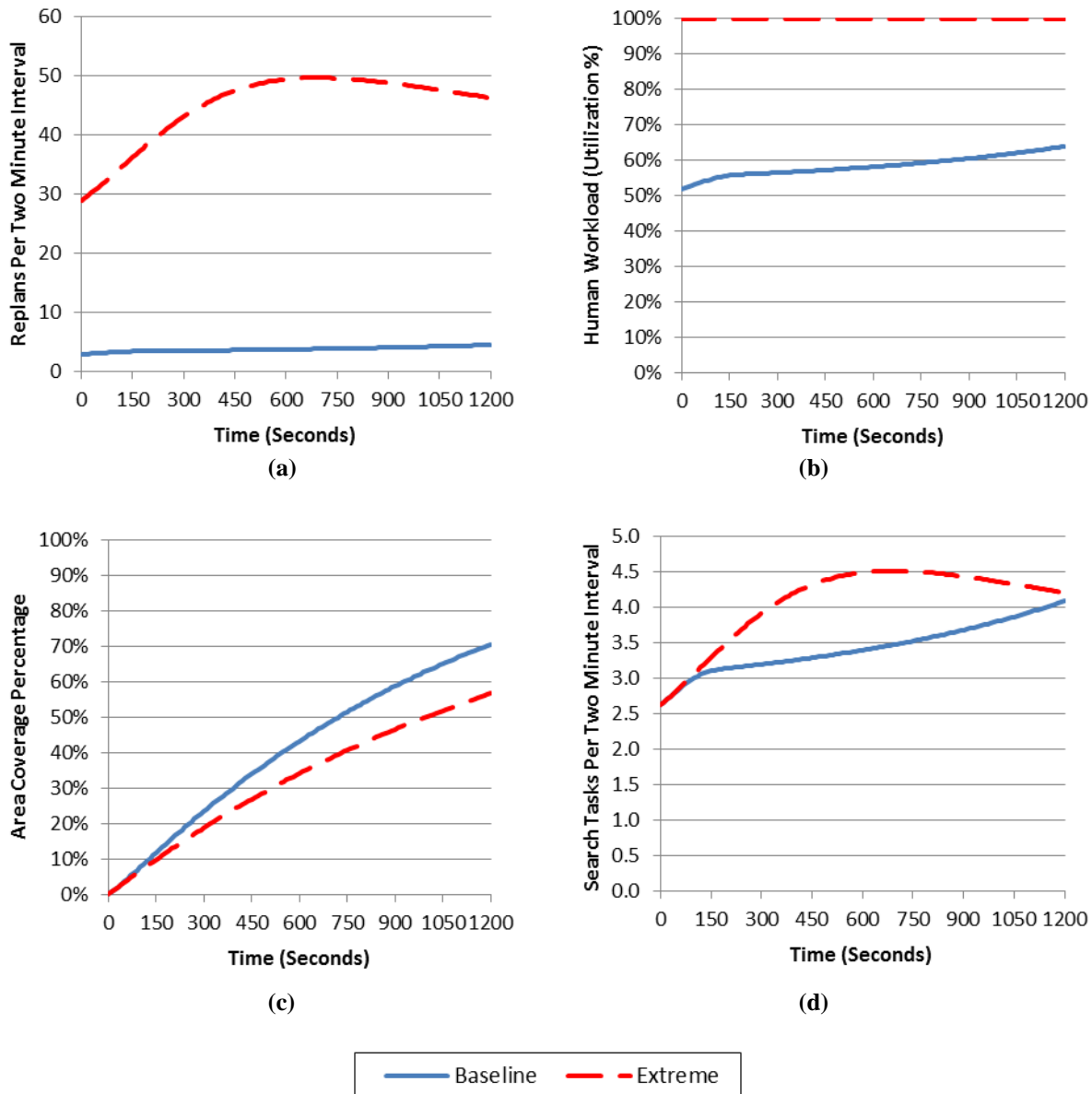
Figure 85. Extreme conditions test: Initial EP set to 10x and 10% of baseline value. (a) PPG. (b) Perceived automation capability. (c) Search task rate. (d) Workload. (e) Area coverage performance.

260

It should be noted that in both the extreme high Initial EP and extreme low Initial EP tests, the operator's expectations adjusted to a more realistic level over time, as shown by the changes in PPG in Figure 85a. Even though the simulated operator initially had a -500% PPG (the system was far outperforming expectations), the simulated operator eventually adjusted its expectations to the point at approximately 450 seconds into the mission where the operator began to have a positive perceived performance gap (expectations higher than perceived performance).

Additionally, Figure 85 shows the behavior of the model in terms of operator interventions, workload, and system performance under these extreme input conditions. When the operator has extremely high expectations of performance, the expected reaction would be to intervene at a high frequency, as shown in Figure 85c. This causes the operator's workload to exceed 70% (Figure 85d), negating many of the positive effects of these interventions due to cognitive overload. Thus, the operator with extremely high expectations of performance does not perform substantially better than the normal condition (Figure 85e) due to the impact of cognitive overload.

When the operator has extremely low expectations of performance, the typical reaction would be to intervene less frequently, as shown in Figure 85c, resulting in a slightly lower workload level (Figure 85d). However, because the operator is not guiding the suboptimal automation frequently enough, system performance suffers compared to the baseline condition (Figure 85e). Overall, the model behaved as expected in terms of operator interventions, workload, and system performance under these extreme input conditions. All model outputs remained within defined limits and the model behavior was reasonable.

In the final test, the Number of Replans Per Search Task was set to 10x the baseline value in the simulation. This caused the Replan Rate to go to extremely high levels compared to the baseline simulation (Figure 86a). However, Human Workload (Figure 86b) went no higher than 100% as designed. The mathematical cap discussed in Section 4.1.3 prevents utilization from exceeding 100%. This artificial means of limiting the workload parameter is necessary for maintaining the validity of model outputs.

Figure 86. Extreme conditions test: Number of Replans Per Search Task set to 10x baseline value.
(a) Replan rate. (b) Workload. (c) Area coverage performance. (d) Search task rate.

Once again, the model demonstrated that under conditions of cognitive overload, system performance would decline (Figure 86c). Additionally, the simulated operator detected that the system was performing poorly and attempted to "work harder" to counteract the poor system performance. This is reflected in the increasing search task rate for the extreme simulation (Figure 86d). As described in more detail in Section 4.1.5, this cycle of attempting to make up for poor performance by intervening more frequently simply leads to further cognitive overload.

Thus, the model behaves appropriately under these extreme workload conditions, capturing the impact of these extreme conditions on system performance and operator interventions. It should be noted that setting the Number of Replans per Search Task to an extremely low value did not have a substantial impact on model behavior. It simply lowered the workload of the operator slightly, but did not impact performance or intervention behavior. In reality, performance should have suffered because the operator was not replanning to assign the newly created search tasks to the team of UVs. Chapter 7 describes the need for future work to develop a more sophisticated model of replanning that also recognizes the impact of replanning on system performance.

Integration Error Test Results

The time step chosen for the CHAS model was 0.125 seconds and the integration method chosen was Euler (the Vensim® simulation software package provides four integration method options: Euler, fourth order Runge-Kutta, second order Runge-Kutta, and Difference). To evaluate whether the model results were sensitive to changes in the time step, the model was run with a time step of 0.0625 seconds, then with a time step of 0.25 seconds. In both cases the overall results of the model simulations did not change, as shown in Figure 87. Similarly, the model was run with a different integration method, fourth order Runge-Kutta integration with a fixed step size, and there were no changes in the results (Figure 88). For all integration error tests, there were differences in the results at the fifth decimal point, however, precision at that level is unnecessary for a model of human decision-making. Thus, it appears that the model results are robust to changes in the time step and integration method.
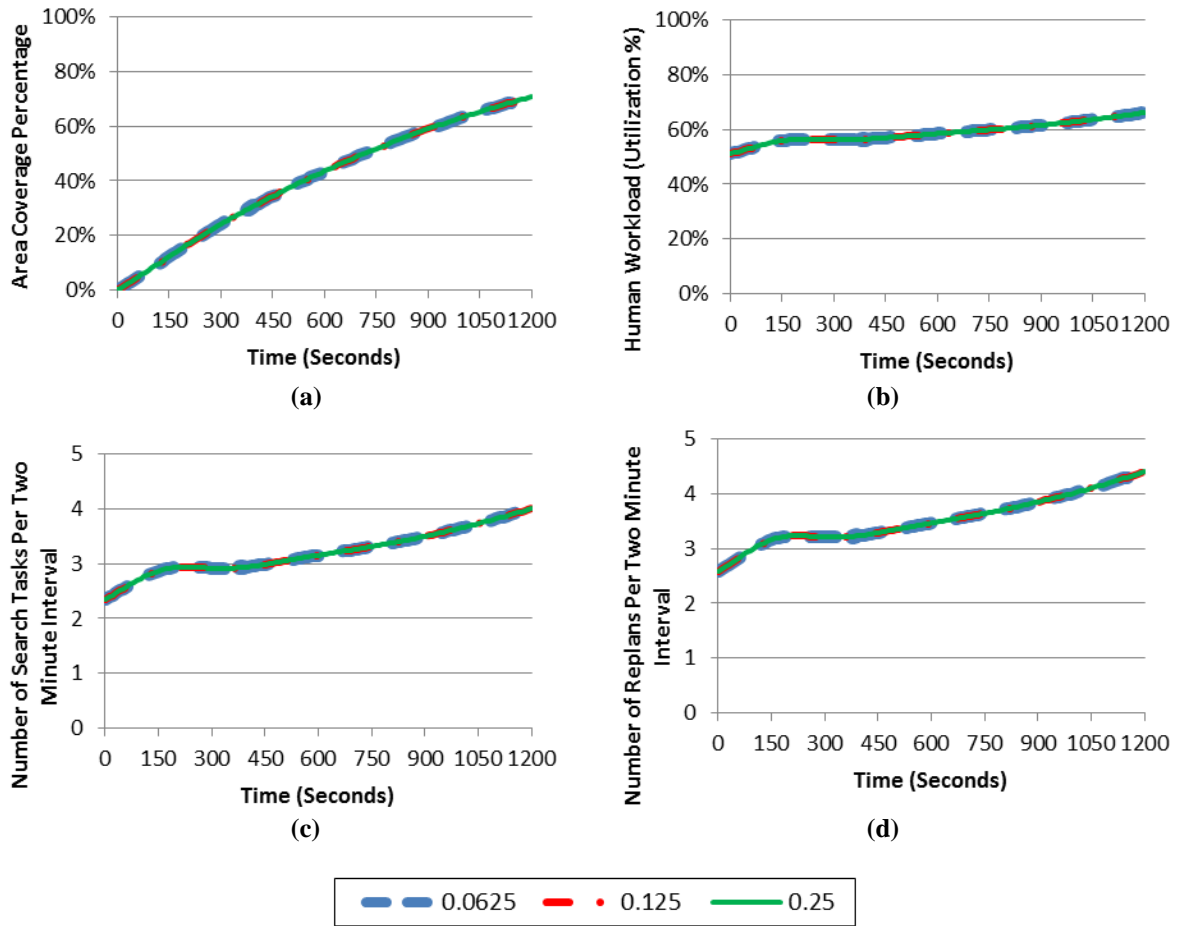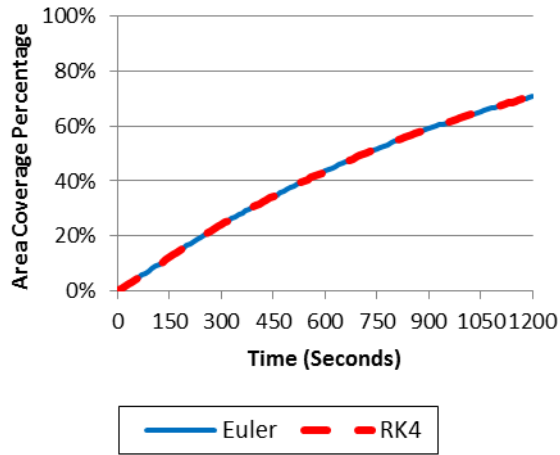
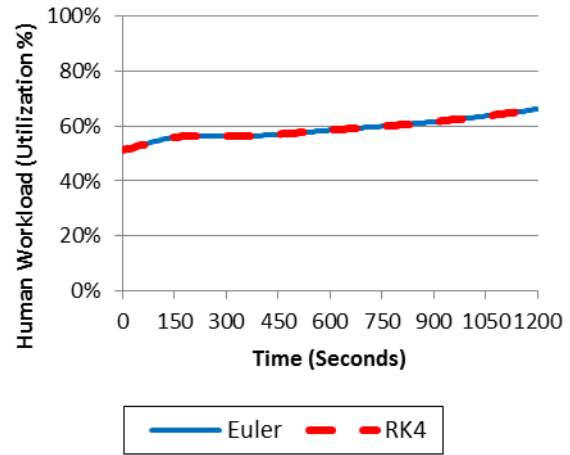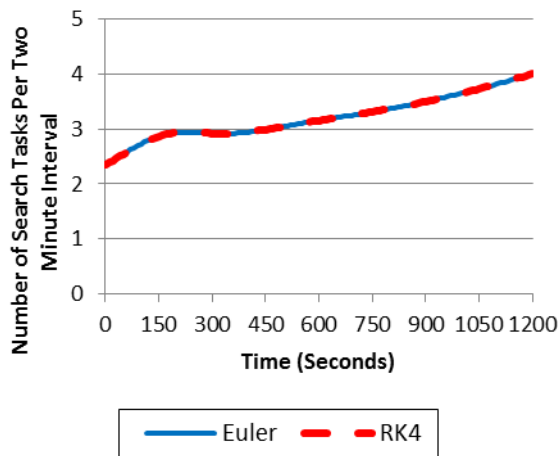**Figure 87. Integration error test with three different time steps. (a) Area coverage. (b) Human workload. (c) Search task rate. (d) Replan rate.**
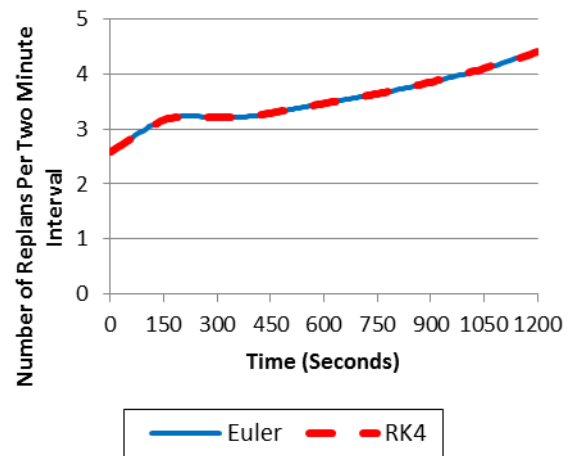
Figure 88. Integration error test with two different integration methods, Euler and fourth order Runge-Kutta. (a) Area coverage. (b) Human workload. (c) Search task rate. (d) Replan rate.

Appendix F: Model Parameters for Original OPS-USERS Experiment

Table 18. All parameter values that remained constant across the three groups: All Missions, Low Performers, and High Performers.

Parameter	Value	Units
Automation Generated Search Speed	2.9	Cells/Second
Base PPG	13	% PPG
Base Search Task Rate	2.85	Search Tasks per Two Minute Interval
Max Human Value Added	2	Cells/Second
Max Search Task Rate	8	Search Tasks per Two Minute Interval
Midpoint Trust	25	% Trust
Min Human Value Added	-1	Cells/Second
Min Search Task Rate	1.9	Search Tasks per Two Minute Interval
Mission Time Length	1200	Seconds
PPG Max Slope	3	Dimensionless
Sampling Interval	120	Seconds
Search Task Creation Time Length	2.8	Seconds
Search Task Rate Max Slope	0.6	Cells/(Second*Search Tasks per Two Minute Interval)
Total Number of Cells	4150	Cells
Trust Max Slope	2	Search Tasks per Two Minute Interval/% Trust

Table 19. All parameter values that were allowed to vary across the three groups: All Missions, Low Performers, and High Performers.

Parameter	All Missions	Low Performers	High Performers	Units
Initial EP	9.8	4	6.25	Cells/Second
Initial Human Trust	88	91	26	% Trust
Initial NST	30	30	30	% Utilization
Length of Time to Replan	6.5	7.7	4.5	Seconds
NST Rate	-0.0002	-0.0002	-0.0002	% Utilization/ Second
Number of Replans per Search Task	1.1	1.2	0.9	Replans/ Search Tasks
Time Horizon for Expected Performance	190	220	220	Seconds
Time to Perceive Present Performance (TPPP)	150	220	210	Seconds
Trust Change Time Constant	410	445	60	Seconds

Appendix G: Individual Mission Replication Testing

While Chapter 4 evaluated the ability of the model to replicate aggregate performance, workload, and intervention actions for groups of operators, another key test of model accuracy is the ability to replicate *individual* operator behavior and performance. Sterman (1989a) performed similar tests by fitting a human decision-making model to individual data gathered using a computer-based simulation of an economy where test subjects deciding how to invest capital to satisfy customer demand. Similarly, Sterman (1989b) fit a different decision-making model to individual trial data using a computer simulation of an industrial production and distribution system, called the "Beer Distribution Game."

The CHAS model was used to replicate the 60 individual missions in the OPS-USERS experimental data set, described in Section 3.2. In order to fit the model to each mission, the optimization feature in the Vensim® simulation software was utilized. The software uses a modified Powell (1964) search to find a local minimum by searching within defined boundaries for each parameter. The optimizer evaluated the fit of the model to experimental data for the following variables: area coverage performance, human workload as measured by utilization, search task rate, replan rate. These four output variables were the only endogenous variables in the CHAS model for which experimental data was available for comparison. Data on the average length of time to replan was used to set the exogenous parameter for this interaction time length.

In the optimization, nine input parameters were allowed to vary: Initial Human Trust (IT), Initial Expected Performance (EP), Trust Change Time Constant (TC), Time to Perceive Present Performance (TPPP), Time Horizon for Expected Performance (THEP), Number of Replans per Search Task (NRST), Length of Time to Replan (LTR), Initial NST, and NST Rate. All other exogenous parameters in the model were either a) known from the testbed and constant for all operators, b) had negligible variation and were kept constant for all operators, or c) estimated from aggregate data in order to define one of the non-linear relationships in the model and thus were kept constant between model runs (see Section 4.1.5 for more details on the selection criteria for these parameters).

The analysis of the results of this individual fitting process was based on the method that Sterman (1989b) utilized to analyze the accuracy of the model fit. The results of the individual

fitting process are presented below. Descriptive statistics of the nine estimated parameters for each mission are shown in Table 20. The R^2 and Root Mean Square Error (RMSE) values of the model fit to the experimental data for four output variables are shown in Table 21. The mean R^2 for the primary performance metric of area coverage was 0.968. The minimum R^2 for area coverage among all 60 missions was 0.838. The other outputs variables had mean R^2 values ranging from 0.160 to 0.335, similar to the aggregate fits achieved in Chapter 4.

Table 20. Descriptive statistics of estimated parameters for model fit to 60 individual missions.

	IT	Initial EP	Trust TC	TPPP	THEP	NRST	LTR	Initial NST	NST Rate
Mean	0.664	4.638	529.274	468.023	552.208	1.031	6.986	0.307	-0.0002
Median	0.794	4.747	300.462	247.557	296.880	1.040	6.649	0.299	-0.0002
Min	0.000	0.263	1.000	1.000	106.824	0.364	2.287	0.105	-0.0004
Max	1.000	10.000	1200.000	1200.000	1200.000	1.890	14.773	0.533	0.0000
St. Dev	0.373	3.493	477.554	467.427	441.927	0.319	2.845	0.087	0.0001
SE	0.048	0.451	61.652	60.345	57.053	0.041	0.367	0.011	0.0000

Table 21. Descriptive statistics of goodness of fit measures for model fit to 60 individual missions.

	Area Coverage		Utilization		Search Task Rate		Replan Rate	
	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE
Mean	0.968	0.082	0.335	0.103	0.299	1.478	0.160	1.083
Median	0.978	0.074	0.354	0.103	0.303	1.386	0.128	1.031
Min	0.838	0.015	0.001	0.033	0.000	0.756	0.000	0.524
Max	0.997	0.238	0.886	0.168	0.896	2.455	0.563	1.847
St. Dev	0.031	0.047	0.243	0.031	0.279	0.438	0.170	0.317
SE	0.004	0.006	0.031	0.004	0.036	0.057	0.022	0.041

As a further test, the model simulation was run with the estimated parameters for each mission. The final area coverage performance by the end of the mission calculated by the simulation was compared to the experimental results for every mission. If the model were perfect, the simulated and experimental performance results would be equal, and regressing the simulated scores on the experimental scores would produce a slope of 1. The actual results for this regression showed a slope of 0.753, with an R^2 value of 0.537. The slope of the relationship was significant ($p < 0.001$) and 2.9 standard errors from unity, indicating a moderate correspondence between the actual and simulated final performance.

Appendix H: Model Parameters for High Task Load Experiment

Table 22. All parameter values that remained constant across the two sets of missions: 30 Second Replan Prompting Interval and 45 Second Replan Prompting Interval.

Parameter	Value	Units
Automation Generated Search Speed	30	Cells/Second
Base PPG	13	% PPG
Base Search Task Rate	6	Search Tasks per Two Minute Interval
Max Human Value Added	2	Cells/Second
Max Search Task Rate	12	Search Tasks per Two Minute Interval
Midpoint Trust	40	% Trust
Min Human Value Added	-1	Cells/Second
Min Search Task Rate	0	Search Tasks per Two Minute Interval
Mission Time Length	1200	Seconds
PPG Max Slope	1.5	Dimensionless
Sampling Interval	120	Seconds
Search Task Creation Time Length	2.8	Seconds
Search Task Rate Max Slope	0.25	Cells/(Second*Search Tasks per Two Minute Interval)
Total Number of Cells	4150	Cells
Trust Max Slope	6	Search Tasks per Two Minute Interval/% Trust

Table 23. All parameter values that were allowed to vary across the two sets of missions: 30 Second Replan Prompting Interval and 45 Second Replan Prompting Interval.

Parameter	30 Second Replan Prompting Interval	45 Second Replan Prompting Interval	Units
Initial EP	33.6	31	Cells/Second
Initial Human Trust	36.5	36.5	% Trust
Initial NST	45	47	% Utilization
Length of Time to Replan	2.9	3.2	Seconds
NST Rate	0	-0.0001	% Utilization/ Second
Replan Rate	5.1	4.5	Replans/ Search Tasks
Time Horizon for Expected Performance	235	60	Seconds
Time to Perceive Present Performance (TPPP)	400	333	Seconds
Trust Change Time Constant	1550	2000	Seconds

272

Appendix I: Model Parameters for USAR Experiment

Table 24. All parameter values that remained constant across the three groups: Low, Medium, and High Teleoperation Groups.

Parameter	Value	Units
Automation Generated Search Speed	0.011	Victims/Second
Base Number of Teleoperations	200	Teleoperations
Base PPG	5	% PPG
Effect of Teleoperations on Workload	0.0013	% Utilization/ Teleoperations
Max Human Value Added	0.01	Victims/Second
Max Number of Teleoperations	800	Teleoperations
Midpoint Trust	45	% Trust
Min Human Value Added	-0.0035	Victims/Second
Min Number of Teleoperations	0	Teleoperations
Mission Time Length	1500	Seconds
Number of Teleoperations Max Slope	0.00125	Victims/(Second *Teleoperations)
PPG Max Slope	1.25	Dimensionless
Total Number of Victims	17	Victims
Trust Max Slope	1.5	Teleoperations/% Trust

Table 25. All parameter values that were allowed to vary across the three groups: Low, Medium, and High Teleoperation Groups.

| Parameter	Low TeleOp Group	Medium TeleOp Group	High TeleOp Group	Units
Initial EP	0.041	0.041	0.041	Victims/Second
Initial Human Trust	80	80	80	% Trust
Time Horizon for Expected Performance	100	100	290	Seconds
Time to Perceive Present Performance (TPPP)	30	30	240	Seconds
Trust Change Time Constant	10000	466	165	Seconds

Appendix J: Monte Carlo Simulation Distributions

The fitted distributions for the four model variables that were selected for use in Monte Carlo simulations are presented in Table 26. These were generated using the EasyFit[©] Software package, with data from the OPS-USERS experiment described in Section 3.2. Graphical representations of the experimental data and fitted distributions are shown in Figure 89.

Table 26. Fitted distributions for Monte Carlo simulation variables.

Model Variable	Lognormal Distribution Parameters
Base Search Task Rate	$\sigma = 0.56146$, $\mu = 0.91572$, min=0.6, max=6
Initial NST	$\sigma = 0.31293$, $\mu = -0.99429$, min=0.2, max=0.55
Length of Time to Replan	$\sigma = 0.77659$, $\mu = 1.5937$, min=0.5, max=25
Number of Replans per Search Task	$\sigma = 0.33953$, $\mu = 0.29366$, min=0.5, max=2.5
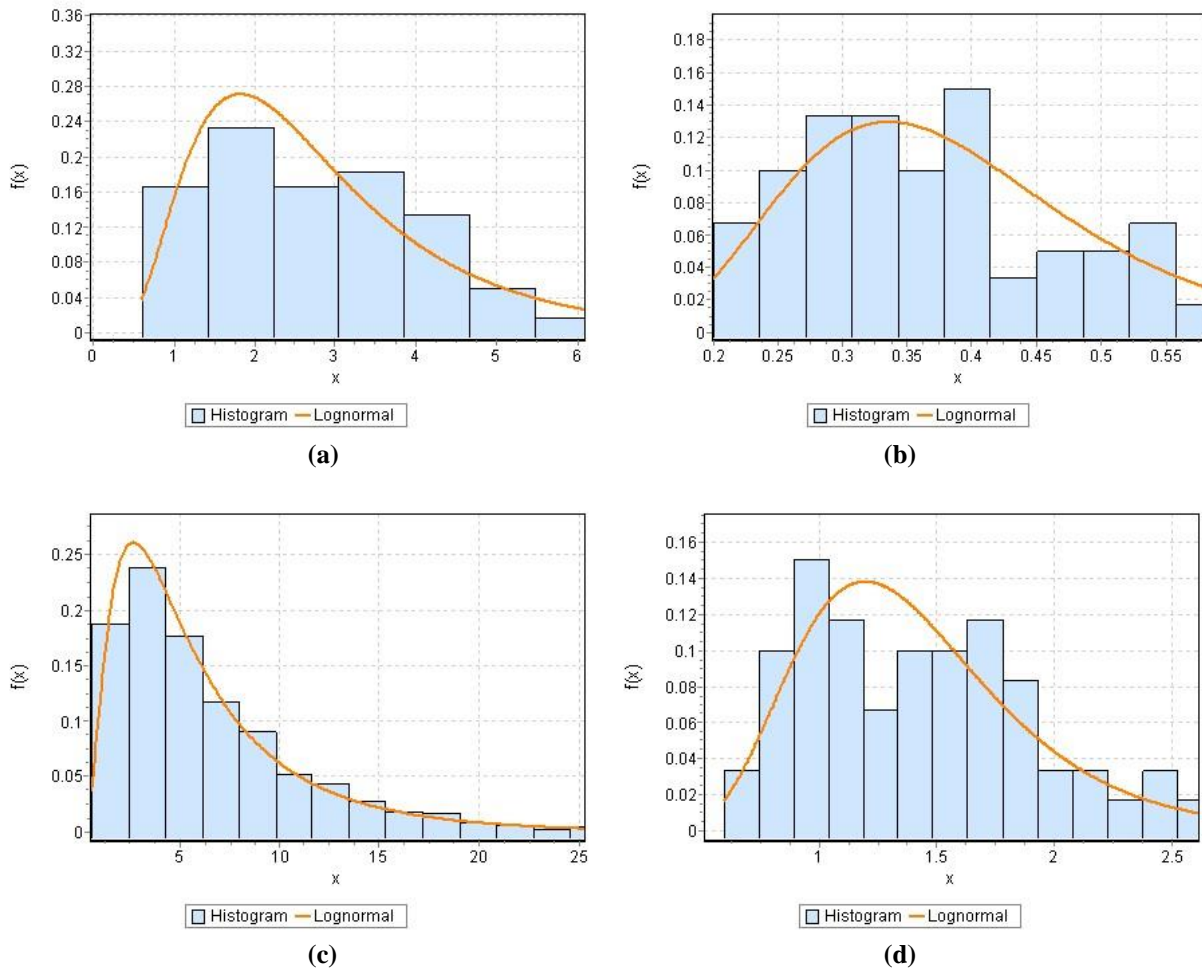
Figure 89. Fitted distributions for (a) Base Search Task Rate. (b) Initial NST. (c) Length of Time to Replan. (d) Number of Replans per Search Task.

275

Appendix K: *A Priori* Priming Passages

Both passages were derived from written comments from a previous OPS-USERS experiment (Clare, Cummings, How, et al., 2012):

Positive Priming:

To give you a better sense of how the Automated Scheduler works, here are some stats and quotes from previous operators who used this system:

- 87% percent of participants indicated that the automated scheduler was fast enough for this dynamic, time-pressured mission.
- "Once you got the hang of it, it was easy to become quickly familiar with the system."
- "I liked that it gave instructions and helped us. It was easy to gain confidence."
- "The system is easy to use and intuitive to work with."
- "The automated scheduler was very fast."
- "There were times when I didn't have to do anything except identify a new target. I usually just accepted the plan created by the automated scheduler."
- "Where can I get one of these? This is fun!"

Negative Priming:

To give you a better sense of how the Automated Scheduler works, here are some stats and quotes from previous operators who used this system:

- 53% percent of participants indicated that they did not agree with the plans created by the automated scheduler and wanted to be able to manually assign tasks.
- "I did not always understand decisions made by the automated scheduler…namely it would not assign tasks…while some vehicles were seemingly idle."
- "The automated scheduler makes some obviously poor decisions…I feel like a lot is hidden from me in the decision making…I felt like I had to trick it into doing things."
- "I wish I could manually assign vehicles to certain spots. It seemed like the automated scheduler wasn't great."
- "I wish that I could have rearranged tasks in the schedule created by the automated scheduler."
- "The automated scheduler would assign a different UAV than I would have picked."
- "The actual [automated scheduler] assigning robots to tasks was a little screwy…Maybe an option to let user designate certain missions to certain vehicles would take full advantage of the superior reasoning capacity of humans."

Appendix L: Consent to Participate Form

CONSENT TO PARTICIPATE IN
NON-BIOMEDICAL RESEARCH

Modeling Human-Automation Collaborative Scheduling of Multiple UVs

You are asked to participate in a research study conducted by Professor Julie Shah Ph.D, and Andrew Clare, S.M., from the Aeronautics & Astronautics Department at the Massachusetts Institute of Technology (M.I.T.). Results from this study will contribute to a Ph.D. Thesis by Andrew Clare. You were selected as a possible participant in this study because the expected population this research will influence is expected to contain men and women between the ages of 18 and 50 with an interest in using computers. You should read the information below, and ask questions about anything you do not understand, before deciding whether or not to participate.

- **PARTICIPATION AND WITHDRAWAL**

Your participation in this study is completely voluntary and you are free to choose whether to be in it or not. If you choose to be in this study, you may subsequently withdraw from it at any time without penalty or consequences of any kind. The investigator may withdraw you from this research if circumstances arise which warrant doing so.

- **PURPOSE OF THE STUDY**

The purpose of this study is to investigate the effect of different system designs and training on overall mission performance when a human operator collaborates with an automated scheduler to control multiple unmanned vehicles.

- **PROCEDURES**

If you volunteer to participate in this study, we would ask you to do the following things:

- Fill out a demographic survey and a Metacognitive Awareness Inventory (Estimated 15 minutes)
- Participate in training to learn a video game-like software environment that will have you control a team of simulated unmanned vehicles. (Estimated 20 minutes)
- Practice on the software environment will be performed until an adequate level of performance is achieved, which will be determined by your demonstration of basic proficiency in operating the unmanned vehicles and replanning the mission. (one 15 minute training session)
- Execute two 20 minute trials consisting of the same tasks as above (Estimated 50 minutes).
- Attend a debriefing to determine your subjective responses and opinion of the software (Estimated 10 minutes).

- After each trial you will be assigned a score for the trial based what percentage of the total area of interest is searched, the number of targets you successfully find, how long they are successfully tracked thereafter, and how well you follow instructions provided by the chat box.
- All testing will take place at MIT in room 35-220.
- Total time: 2 hours, depending on skill level.

• POTENTIAL RISKS AND DISCOMFORTS

There are no anticipated physical or psychological risks.

• POTENTIAL BENEFITS

While you will not directly benefit from this study, the results from this study will assist in the design of interfaces for human/unmanned vehicle systems.

• PAYMENT FOR PARTICIPATION

You will be paid $20 for your participation in this study, which will be paid upon completion of your debrief. Should you elect to withdraw during the study, you will be compensated for your time spent in the study. The subject with the best performance will be given a reward of a $100 Amazon Gift Certificate.

• CONFIDENTIALITY

Any information that is obtained in connection with this study and that can be identified with you will remain confidential and will be disclosed only with your permission or as required by law. You will be assigned a subject number which will be used on all related documents to include databases, summaries of results, etc. Only one master list of subject names and numbers will exist that will remain only in the custody of Professor Shah.

• IDENTIFICATION OF INVESTIGATORS

If you have any questions or concerns about the research, please feel free to contact the Principal Investigator, Julie Shah, at (617) 324-4879, e-mail, arnoldj@mit.edu, and her address is 77 Massachusetts Avenue, Room 33-305, Cambridge, MA 02139. The graduate student investigator is Andrew Clare. He may be contacted at (617) 253-0993 or via email at aclare@mit.edu.

• EMERGENCY CARE AND COMPENSATION FOR INJURY

If you feel you have suffered an injury, which may include emotional trauma, as a result of participating in this study, please contact the person in charge of the study as soon as possible.

In the event you suffer such an injury, M.I.T. may provide itself, or arrange for the provision of, emergency transport or medical treatment, including emergency treatment and follow-up care, as needed, or reimbursement for such medical services. M.I.T. does not provide any other form of compensation for injury. In any case, neither the offer to provide medical assistance, nor the actual provision of medical services shall be considered an admission of fault or acceptance of liability. Questions regarding this policy may be directed to MIT's Insurance Office, (617) 253-2823. Your insurance carrier may be billed for the cost of emergency transport or medical treatment, if such services are determined not to be directly related to your participation in this study.

- **RIGHTS OF RESEARCH SUBJECTS**

You are not waiving any legal claims, rights or remedies because of your participation in this research study. If you feel you have been treated unfairly, or you have questions regarding your rights as a research subject, you may contact the Chairman of the Committee on the Use of Humans as Experimental Subjects, M.I.T., Room E25-143B, 77 Massachusetts Ave, Cambridge, MA 02139, phone 1-617-253 6787.

| SIGNATURE OF RESEARCH SUBJECT OR LEGAL REPRESENTATIVE |

I understand the procedures described above. My questions have been answered to my satisfaction, and I agree to participate in this study. I have been given a copy of this form.

_____

Name of Subject

_____

Name of Legal Representative (if applicable)

_____   _____

Signature of Subject or Legal Representative   Date

| SIGNATURE OF INVESTIGATOR |

In my judgment the subject is voluntarily and knowingly giving informed consent and possesses the legal capacity to give informed consent to participate in this research study.

_____   _____

Signature of Investigator   Date

Appendix M: Demographic Survey

Pre-experiment Survey

Page 1

1. Subject number:_____

2. Age:_____

3. Gender: _____

4. Occupation:_____

    if student, (circle one):   *Undergrad*   *Masters*   *PhD*

5. Military experience (circle one):   *No*   *Yes*

    If yes, which branch:_____

        Years of service:_____

6. Give an overall rating of your past two nights of sleep.

    Poor   *Fair*   *Good*   *Great*

7. On average, how much TV do you watch daily?

    Never watch TV   Infrequently watch TV   About 1 hour   About 2 hours   More than 2 hours

8. How often do you play computer games?

    Rarely play games   Play games once a month   Weekly gamer   A few times a week gamer   Daily gamer

    Types of games played:

9. Rate your comfort level with using computers.

    Not comfortable   Somewhat comfortable   Comfortable   Very Comfortable

10. What is your perception toward unmanned vehicles?

    Intense dislike   Dislike   Neutral   Like   Really Like

Metacognitive Awareness Inventory

Pre-experiment Survey

Page 2

Subject number:_____

	Strongly Disagree/ Never		Neutral/ Sometimes		Strongly Agree/ Always
1. I ask myself periodically if I am meeting my goals.	1	2	3	4	5
2. I consider several alternatives to a problem before I answer.	1	2	3	4	5
3. I try to use strategies that have worked in the past.	1	2	3	4	5
4. I pace myself while learning in order to have enough time.	1	2	3	4	5
5. I understand my intellectual strengths and weaknesses.	1	2	3	4	5
6. I think about what I really need to learn before I begin a task.	1	2	3	4	5
7. I know how well I did once I finish a test.	1	2	3	4	5
8. I set specific goals before I begin a task.	1	2	3	4	5
9. I slow down when I encounter important information.	1	2	3	4	5
10. I know what kind of information is most important to learn.	1	2	3	4	5
11. I ask myself if I have considered all options when solving a problem.	1	2	3	4	5
12. I am good at organizing information.	1	2	3	4	5
13. I consciously focus my attention on important information.	1	2	3	4	5
14. I have a specific purpose for each strategy I use.	1	2	3	4	5
15. I learn best when I know something about the topic.	1	2	3	4	5
16. I know what the teacher expects me to learn.	1	2	3	4	5
17. I am good at remembering information.	1	2	3	4	5
18. I use different learning strategies depending on the situation.	1	2	3	4	5
19. I ask myself if there was an easier way to do things after I finish a task.	1	2	3	4	5
20. I have control over how well I learn.	1	2	3	4	5
21. I periodically review to help me understand important relationships.	1	2	3	4	5
22. I ask myself questions about the material before I begin.	1	2	3	4	5
23. I think of several ways to solve a problem and choose the best one.	1	2	3	4	5
24. I summarize what I've learned after I finish.	1	2	3	4	5
25. I ask others for help when I don't understand something.	1	2	3	4	5
26. I can motivate myself to learn when I need to.	1	2	3	4	5
27. I am aware of what strategies I use when I study.	1	2	3	4	5
28. I find myself analyzing the usefulness of strategies while I study.	1	2	3	4	5
29. I use my intellectual strengths to compensate for my weaknesses.	1	2	3	4	5
30. I focus on the meaning and significance of new information.	1	2	3	4	5
31. I create my own examples to make information more meaningful.	1	2	3	4	5
32. I am a good judge of how well I understand something.	1	2	3	4	5
33. I find myself using helpful learning strategies automatically.	1	2	3	4	5
34. I find myself pausing regularly to check my comprehension.	1	2	3	4	5
35. I know when each strategy I use will be most effective.	1	2	3	4	5
36. I ask myself how well I accomplish my goals once I'm finished.	1	2	3	4	5
37. I draw pictures or diagrams to help me understand while learning.	1	2	3	4	5
38. I ask myself if I have considered all options after I solve a problem.	1	2	3	4	5
39. I try to translate new information into my own words.	1	2	3	4	5
40. I change strategies when I fail to understand.	1	2	3	4	5
41. I use the organizational structure of the text to help me learn.	1	2	3	4	5
42. I read instructions carefully before I begin a task.	1	2	3	4	5
43. I ask myself if what I'm reading is related to what I already know.	1	2	3	4	5
44. I reevaluate my assumptions when I get confused.	1	2	3	4	5
45. I organize my time to best accomplish my goals.	1	2	3	4	5
46. I learn more when I am interested in the topic.	1	2	3	4	5
47. I try to break studying down into smaller steps.	1	2	3	4	5
48. I focus on overall meaning rather than specifics.	1	2	3	4	5
49. I ask myself questions about how well I am doing while I am learning something new.	1	2	3	4	5
50. I ask myself if I learned as much as I could have once I finish a task.	1	2	3	4	5
51. I stop and go back over new information that is not clear.	1	2	3	4	5
52. I stop and reread when I get confused.	1	2	3	4	5

Pre-experiment Survey

Page 3

1. Subject number:_____

2. How confident were you about your performance?

   Not Confident   Somewhat Confident   Confident   Very Confident   Extremely Confident

   Comments:

3. How much do you trust the Automated Scheduler on a scale from 1-7? (please circle one)

   Not at all   Absolute Trust
   1   2   3   4   5   6   7

4. How busy did you feel during the practice mission?

   Extremely Busy   Busy   Not Busy   Idle

5. Do you understand how to create search tasks?

   No   Somewhat   Yes

6. Do you understand how to use the target identification window?

   No   Somewhat   Yes

7. Do you understand how to approve a weapon launch on hostile targets?

   No   Somewhat   Yes

8. Do you understand how to use the Schedule Comparison Tool (SCT)?

   No   Somewhat   Yes

9. Do you understand that you must accept a plan in order for the unmanned vehicles to perform new search, track and destroy tasks?

   No   Somewhat   Yes

10. Do you understand how to modify the objective function of the automated planner?

   No   Somewhat   Yes

Now is the time to ask the experiment administrator any questions you have about the mission or interface.

Appendix N: Demographic Descriptive Statistics

Category	N	Min	Max	Mean	Std. Dev.
Age (years)	48	18	32	23.08	3.842
Rating of past 2 nights of sleep (1-4)	48	1	4	2.56	0.848
Rating of TV watching (1-5)	48	1	5	2.27	1.106
Rating of gaming experience (1-5)	48	1	5	2.23	1.309
Rating of comfort level with computers (1-4)	48	2	4	3.54	0.617
Rating of perception of unmanned vehicles (1-5)	48	2	5	3.94	0.783
Occupation (Student/Other)	Undergraduate: 17 Masters: 14 Ph.D: 15 Non-student: 2	-	-	-	-
Military experience (Y/N)	5/43	-	-	-	-
Gender (M/F)	35/13	-	-	-	-
Metacognitive Awareness Inventory (MAI) Score	48	141	229	189.71	18.96

Appendix O: Experiment Legend

Legend

UxV Symbols

Weaponized Unmanned Aerial Vehicle (WUAV)
- Primary Mission: Detect and Destroy Hostiles

Unmanned Aerial Vehicles 2 & 3 (UAVs)
- Primary Mission: Search and Track

Unmanned Surface Vehicle 1 (USV)
- Primary Mission: Search and Track

Base – Refueling Location

Search Task Symbols

High Priority	Medium Priority	Low Priority
3D	4E	F

Loiter Symbols

High Priority	Medium Priority	Low Priority
D	C	B

Target Symbols

Hostile	Unknown	Friendly
wD	2B	G

Legend

Lost Target

Destroyed Hostile Target

Performance Plot: Area Coverage

Red is the
Baseline Performance
based on how well previous operators have done on this mission

Blue is your
Actual Performance
and sometimes there is a delay in receiving this information
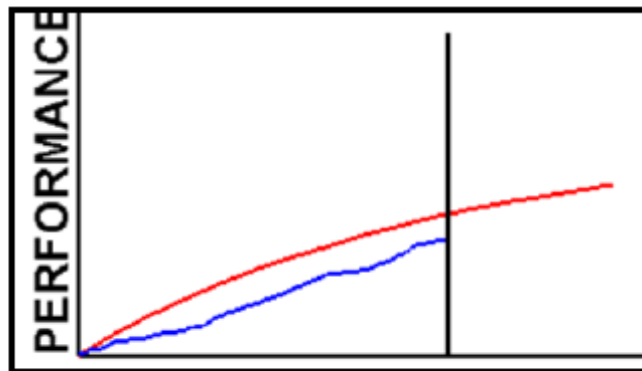
Current Time

Red above Blue, you have work to do.

Blue above Red, you're getting ahead!

Your Mission Score

Overall Mission Score will be calculated by:

- % Area Covered by Mission End
- % Targets Found
- % Time Targets Tracked
- % Hostile Targets Destroyed
- Chat Box Response Time and Accuracy

290

Appendix P: Rules of Engagement

The following Rules of Engagement were sent through the Chat Window to the operator at the specified times:

- START: *Cover as much area as possible to find new targets. Tracking found targets is low priority. Do not destroy any hostiles*.

- FIVE MINUTES: *Conduct search tasks in SE and SW Quadrants. 2nd priority: Track all targets previously found. Do not destroy any hostiles*.

- TEN MINUTES: *Track all targets closely - it is important not to lose any targets! 2nd priority: conserve fuel. 3rd priority: destroy hostile targets*.

- FIFTEEN MINUTES: *All Hostile Targets are now high priority - destroy all hostiles!*

292

Appendix Q: Experiment PowerPoint Tutorials

Loiter Symbols

The loiter symbol for the WUAV resembles a stop sign.
The color indicates priority level.

High Priority   **Medium Priority**   **Low Priority**

D   C   B

The Weaponized UAV does not attend to search tasks. The auto-planner will not schedule the WUAV to track targets. The mission of the WUAV is to **detect and destroy hostile targets**. The WUAV can be sent to loiter, or hover over a particular position, while waiting to destroy hostile targets. The WUAV can only search the map (increasing the coverage area) on the route to a loiter task. The WUAV will not search on its own like the other UxVs.

Target Symbols

Hostile   **Unknown**   **Friendly**

WD   2B   G

The UxVs must **periodically track**, or revisit, the targets that have been found. The Weaponized UAV must **destroy hostile targets**.

- Red diamonds are hostile targets
- Yellow clovers are unknown targets
- Blue rectangles are friendly and are not tracked
- The letter on the right identifies the target
- The character on the left indicates which UxV is assigned to the target

For example, according to the hostile symbol on the left, the Weaponized UAV is assigned to destroy hostile target D. Next, the center symbol shows that UAV 2 will track Unknown Target B. UAV 2 will travel to the location where this target symbol is positioned on the map and begin following the target. If UAV 2 has another task to perform or must go back to base to refuel, the algorithm will calculate an estimated new position for the target based on the target's last known position and velocity. Friendly target G, on the right, is not assigned because friendlies are not to be tracked.

Target Symbol Priority Levels

D   2B   G

C   3E   2A   F
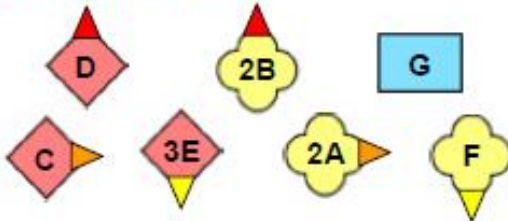
Flags designate target priority level.

- Red vertical flag on top of the target symbol specifies high priority
- Orange horizontal flag beside the target symbol specifies medium priority
- Yellow downward flag below the target symbol specifies low priority
- Friendlies do not have a priority level flag because they do not need to be tracked
  - Only hostile and unknown targets are tracked

Overview of the Map Display

- Getting Started
- Parts of the Map Display
- Selecting a UxV
- Creating or Editing a Search Task
- Selecting an Existing Search Task
- Chat Message Box
- Mini Map
- Identifying a Target
- Selecting a Target
- Lost Targets
- Approving Weapon Launch
- Timeline
- Coverage Area Overlay
- Performance Plot
- Replanning
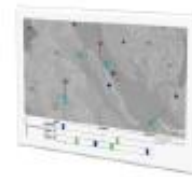
Getting Started

Click the **green start button** in the center of the interface to begin the scenario.

Map Display

The Map Display shows **Symbols** for UxVs, Search Tasks, Loiter Tasks, & Targets

Mini Map

Performance Plot

Chat Message Box

UxV Task Timeline

294
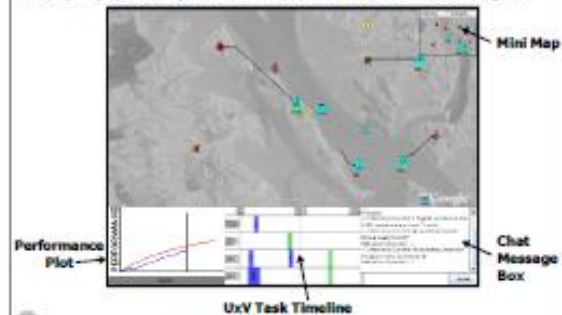
Selecting a UxV

To select a UxV, left click the UxV's symbol.

UxV's **Obstacles** appear as a dark overlay where the UxV cannot go

UxV's **Sensor range** appears in lime green

UxV's **Timeline** is highlighted in light yellow
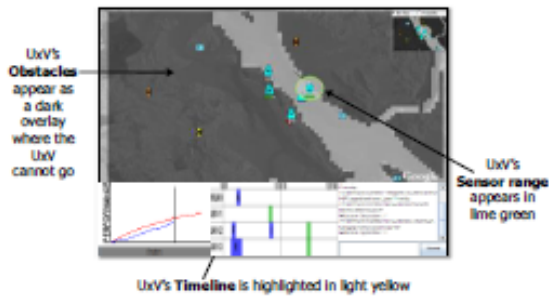
Creating or Editing a Search Task

Right clicking a location on the Map View brings up the Search Task Creation Window. Right click an existing Search Task to edit it using the same window.

Drag the ends of the timeline to **specify a window of opportunity** for when you want the task to begin.

Use this same window to **create or edit a WUAV Loiter Task**. Select the WUAV Loiter checkbox.

Designate the priority level by clicking the appropriate button. The priority is used by the auto-planner to add the task to the schedule.
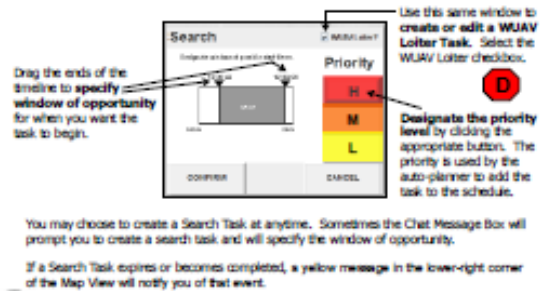
You may choose to create a Search Task at anytime. Sometimes the Chat Message Box will prompt you to create a search task and will specify the window of opportunity.

If a Search Task expires or becomes completed, a yellow message in the lower-right corner of the Map View will notify you of that event.

Selecting an existing Search Task

Click on an existing Search Task Symbol to select it.

Task symbol outlined in lime green when selected

Task **outlined in the timeline** of the UxV assigned to perform the task (if assigned)
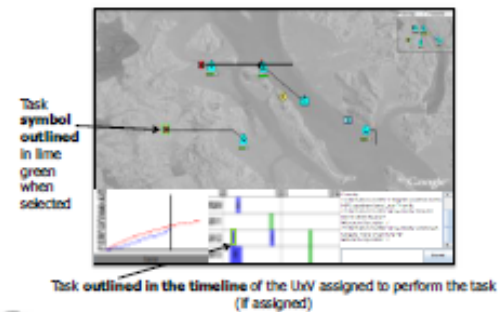
Chat Box: Intelligence Information

Command Center sends intelligence information through chat box instant messaging. The Chat Box gives important information dictating priority levels for targets.

message input window

Send button

When a message comes in, you will hear a tone and the black outline of the box will blink. You can **click in the Chat Box** to stop it from blinking.

Some messages require responses. Click in the **message input window**. Type your message. Then hit enter or click the **send button**. One-word answers or numbers are preferred.

Chat Box: Rules of Engagement

Command Center also sends changes to the **Rules of Engagement** through chat box instant messaging. These Rules of Engagement (ROE) dictate what is most important for you to accomplish. You should adjust your strategy for managing and tasking your vehicles whenever you receive new ROE. Your performance score depends on how well you execute the mission described by the ROE. If you receive new ROE, the old ROE no longer apply.
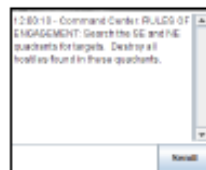
Mini Map

The upper right-hand corner of the Map Display is equipped with a Mini Map that shows the symbols for UxV's, search and loiter tasks, and targets as they appear on the map. Since you can zoom in on the big map using roller wheel on your mouse, it is convenient to glance at the Mini Map for a quick view of the overall picture. The Mini Map feature can be turned on or off by checking or un-checking the Mini Map box above the Mini Map itself. The choice to use this feature is yours.

Mini Map
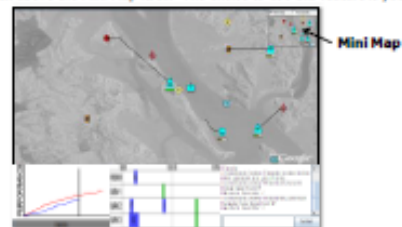
Identifying a Target

In real search & track missions, when a target is found, an operator must identify what type of target has been found. To simulate this, a Target ID Window pops up automatically when a target is found. The Chat Box will provide information on target priority level. You can drag the Target ID Window around the screen.
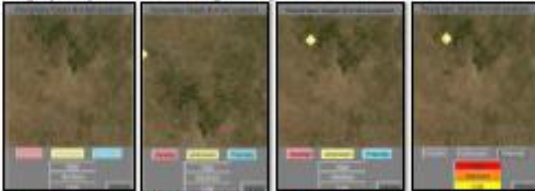
| Initially target may not be visible. | Click and drag over the area to pan for the target symbol. | Click the appropriate target designation button. | Click a priority level button. |

Unknown targets must first be marked as "Unknown," but the designation can be edited afterwards as more information arrives from the Chat Box.

Target ID Video—Click on the Picture to Watch

Selecting a Target
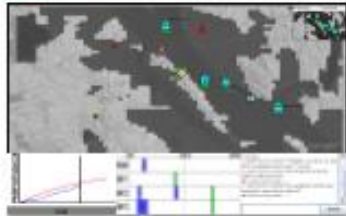
Left clicking a target will cause the following:
- Target's obstacles appear (targets can be land or water vehicles)
- Target's likely position is shown by blue concentric probability rings (if the target has not been visited in a while)
- Target highlighted in the timeline of the UxV assigned to perform it (if assigned)

Right clicking a Target will cause the following:
- Target ID Window appears for editing designation or priority level

Changing Target Designation Video

Right clicking a target allows you to edit the target designation and priority level. The video below shows how to change an unknown target into a hostile, high priority target. The unknown target has been right clicked to bring up the Target ID Window. **Click the picture to watch.**

Target Concentric Rings for possible new location
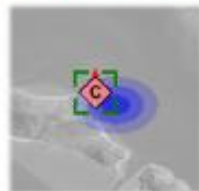
Blue Concentric rings appear automatically for targets that have not been visited in a while that might soon become lost targets. The blue concentric rings shown next to the target symbol represent the algorithm's best estimate of where the target might be. The darker the blue, the more probable the position of the target. A UxV should be assigned a search task near this location ASAP, lest the target become lost.

Lost Targets

Targets become lost if UxVs do not track them for a while.
Lost targets appear dimmer. The probability distribution (blue concentric rings) appear before the target becomes lost and dims.

Compare the gray color of lost Target L on the left to Target A on the right that is not lost.

Lost Target	Target (not lost)
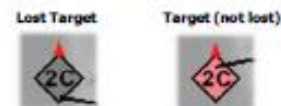
It can take anywhere from 30 to 120 seconds for a target to become lost, depending on target's speed and vehicle type. Revisiting targets more often prevents targets from becoming lost.

A lost target is removed from the map and the schedule queue after the UxVs have revisited the lost icon five times with no success of finding the lost target.

296

Approving Weapon Launch

Weapon Launch Approval Window

Destroyed Hostile Target

Destroyed targets appear as black symbols on the Map View.

Operator approval must be given before the WUAV is allowed to destroy a hostile target. The Missile Launch Approval Window pops up automatically when the WUAV sights the hostile target for destruction and a second UxV has the hostile in its sights as the "eyes on." Pan the screen for a direct view of the target, and click the red **Approve Launch** button to destroy the target.

Timeline
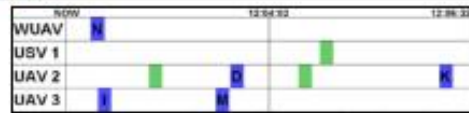
The timeline gives temporal information for each UxV for the next **five minutes into the future** indicated in the format hours:minutes:seconds.

Green bars in the Timeline indicate times of refueling, and blue bars indicate times performing a task. The letter of the task (whether Search Task or Target Track Task) appears in the blue bar. Each UxV is limited to two task assignments at a time. White space indicates idle time or time traveling between tasks. The timeline marches forward as time goes on.

The timeline will begin to gray-out, indicating the end of the mission. The mission end time occurs at the border from white to gray.

Coverage Area Overlay

The Probability Distribution Overlay can be turned on/off by selecting the **Coverage Area checkbox**. The choice to use this feature is yours.
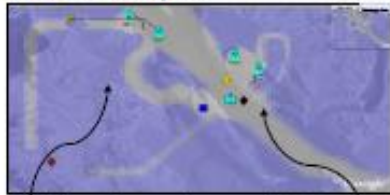
The **periwinkle blue** coloring over the map shows the area that has not been visited recently. These blue areas are more likely to have hidden targets because a UxV has not visited there yet.

The typical **gray** coloring shows a path that was just visited by a UxV, and this coverage area most likely does not contain hidden targets.

Performance Plot: Area Coverage

The Performance Plot shows the amount of **area coverage** you have achieved throughout the mission in comparison to a baseline showing how well previous users of the system have done on this particular mission. The plot marches forward as time goes on, much like the Timeline.

Red is the **Baseline Performance** based on how well previous operators have done on this mission

Blue is your **Actual Performance** and sometimes there is a delay in receiving this information

Red above **Blue**, you have work to do.

Blue above **Red**, you're getting ahead!

Current Time
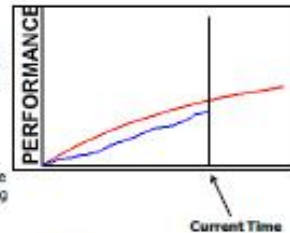
Replan

The Replan button in the lower left corner of the Map Display will turn green when a new plan is available from the auto-planner. You will hear the auditory "Replan" alert. **Click to hear the "Replan" Alert** ➡
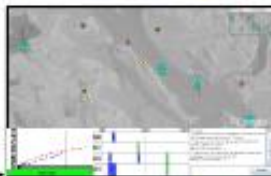
As the Replan button turns green and you hear the alert, a new proposed schedule is available that requires you to hit the replan button to see it.

The auto-planner prompts you to replan when it has generated a better schedule for the UxVs. New targets and tasks in the scenario often result in a new proposed schedule.

Overview of the Schedule Comparison Tool

- Schedule Comparison Tool Display
- Configural Display
- Number on Bars Feature
- Multi-Color Bars Feature
- Assigning Unassigned Tasks
- Nothing to Assign
- Selecting a Task
- Switching Schedules

Schedule Comparison Tool (SCT)

Once you click the replan button, the Schedule Comparison Tool (SCT) appears, showing a performance overview for three schedules: the **Current Schedule** (gray), the **Working Schedule** (blue), and the **Proposed Schedule** given by the automation (green).

After using the SCT, you must click **Accept** in order to implement the working schedule you created and to return to the Map Display.

You may always **Cancel** and return to the Map Display. However, the proposed plan will not be communicated to the UxVs.

In the SCT interface, the timeline has been grayed out to indicate that it is the timeline for the "Current Schedule" and will only correspond to the "Current Schedule."

You may click the Replan button and go to the SCT anytime you like, even without being prompted.
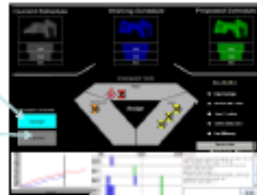
Changing the Priorities of the Automated Planner

Change the priorities of the Automated Planner using the "Plan Priorities" checkboxes in the SCT. You can choose any combination of the options by clicking in the checkboxes.
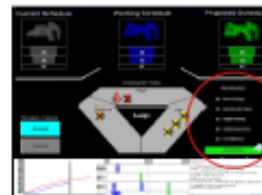
After changing any of the Priorities, you must click **Recalculate** to submit the Priorities to the Automated Planner and to view the new optimal plan

If you click **Accept** to exit the SCT, your Plan Priorities will be saved and you will be prompted to replan when a plan is created that is "better," based on the Priorities that were saved. If you click Cancel, the Priorities revert to the last saved Priorities from when you accepted a plan in the SCT.

Changing the Priorities of the Automated Planner

Plan Priorities
- Area Coverage
- Search/Loiter Tasks
- Target Tracking
- Hostile Destruction
- Fuel Efficiency

Recalculate
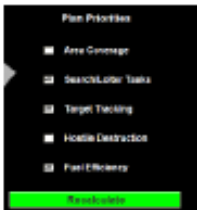
Area Coverage: When selected, the vehicles will cover as much area as possible, neglecting search tasks and tracking targets in order to optimally search the area.

Search/Loiter Tasks: When selected, the vehicles will perform search tasks that you have created and the WUAV will go to specific loiter points that you have created.

Target Tracking: When selected, the vehicles will track already found targets. Hostiles must be tracked by a UxV before they can be destroyed.

Hostile Destruction: Once a hostile target is found and tracked by one of the regular UxVs, it is eligible to be destroyed by the WUAV – the WUAV will only be tasked to destroy these hostiles if this Plan Priority is selected

Fuel Efficiency: When selected, the vehicles will travel more slowly, but also burn fuel more slowly and not have to refuel as often.

Assigning Unassigned Tasks

Assign a task by dragging it into center "Assign" area. Doing this queries the autonomous planner to see if the task can be assigned or not. If the task does not pop back out, the auto-planner was able to schedule the task. Sometimes a different task pops out because the auto-planner had to un-assign that task in order for the task you dragged in to become assigned. This will be reflected in the hierarchical bars as tasks pop in and out of the schedule query "Assign" area.
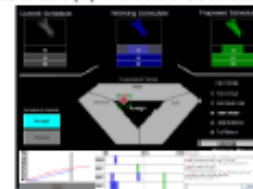
After you finish assigning tasks and working with the schedule, click Accept to implement the plan (or Cancel), and you will return to the Map Display.

Assigning Unassigned Tasks Video—Click the Picture to Watch
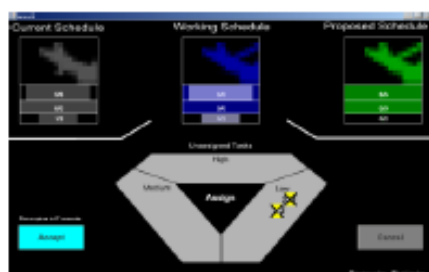
Configural Display

The **top rectangle** represents the "map" area that will be covered for a given schedule. The more colorful the area, the better searched it is.

The **bottom hierarchical ladder** shows the percentages of high, medium, and low priority tasks to be completed for a given schedule. The more color-filled a rectangle is, the more of that task-priority is being done.

High →
Medium →
Low →
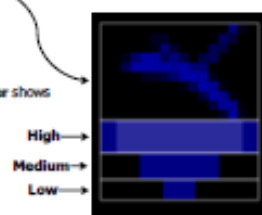
When a task is assigned, the corresponding ladder rung rectangle changes size in a "**ghosting effect**" to visually show what has changed. The darker color is the new size, whereas the lighter overlay is the old size.

298

Ghosting Effect Video—Pay Attention to the Working Schedule
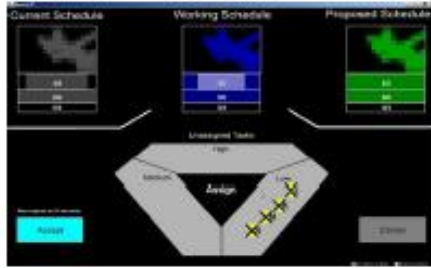
Configural Display
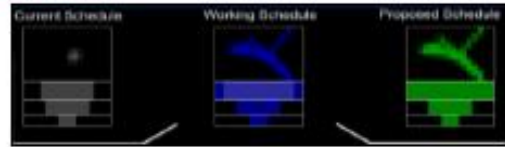
Area Covered
Comparing these three schedules in the above example: both "Working" and "Proposed" cover more area than "Current."

Tasks Completed
Comparing these three schedules: both "Working" and "Proposed" perform more high priority tasks than "Current." "Working" has just been altered (as shown by the ghosting change in the high priority ladder rung).

"Working" starts out identical to "Proposed" and changes as you work on it.

Configural Display "Numbers on Bars" Feature

The ratio of tasks assigned to total tasks will show up on the bars of the hierarchical ladder when the optional "Numbers on Bars" checkbox is selected. For example, on the high priority bar, 3 of the 5 high priority tasks are assigned.
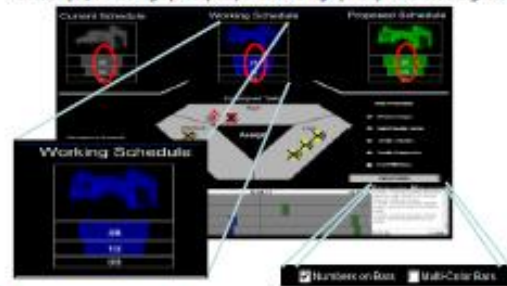
Configural Display "Multi-Color Bars" Feature

The priority level of the bars on the hierarchical ladder will show up for quick priority recognition when the "Multi-Color Bars" checkbox is selected. It is your choice to use this feature.
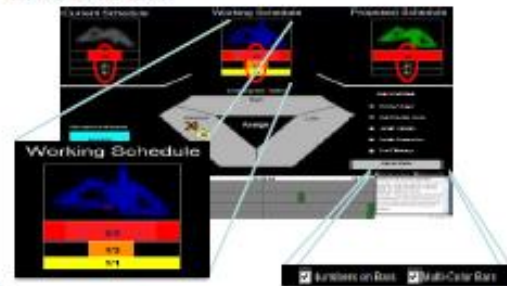
Nothing to Assign
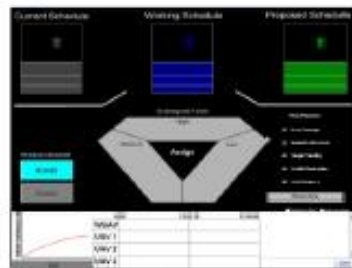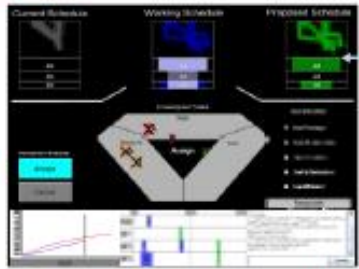
Sometimes the Auto-planner is able to assign all tasks. The "Unassigned Tasks" area will be empty. Click **Accept** to proceed with the completely assigned schedule generated by the auto-planner. If you do not choose Accept, the UxVs will not receive the communication of tasks they need to perform. But, you still have the option to Cancel.

Selected Task shown on configural display
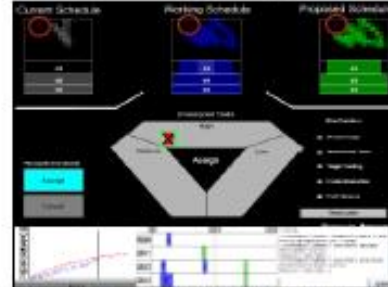
The red X appearing in the "map" of each Configural Display corresponds to the location of the currently **selected task**.

299

Switching Schedules

Click to Switch

Tasks viewed in the unassigned central area always correspond to the "Working" Schedule. To view unassigned tasks for either the "Current" or "Proposed" Schedules, clicking on each schedule's configural display will set the "Working" Schedule as equivalent to the selected schedule (erasing the old "Working").
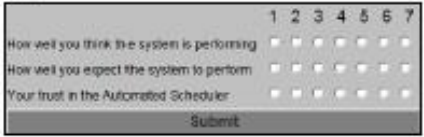
43/47

Survey Window

Every 2 minutes throughout all of your missions, this window will appear in the bottom left corner of the Map Display. It asks you to rate three questions:
- Rate how well you think the **system is performing** right at this moment
  - (1=Extremely Poor, 4=OK, 7=Extremely Well)
- Rate how well you **expect** the system to be performing at this moment
  - (1=Extremely Poor, 4=OK, 7=Extremely Well)
- Rate your **trust in the Automated Scheduler** that you work with in the SCT
  - (1=No Trust, 4=Neutral, 7=Absolute Trust)

Click on one radio button for each question, then click "Submit" and the window will disappear.

44/47

Main parts of your role

Monitoring the Map Display

Identifying Targets

Creating Search Tasks

Using the Schedule Comparison Tool

Approving Weapons Launch

45/47

Your Mission Score

Overall Mission Score will be calculated by:
- Percentage of Area Covered by Mission End
- Number of Targets Found
- Percentage of Targets Tracked (ratio of the time each target is tracked to the total time after the target is found)
- Number of Hostile Targets Destroyed
- Chat Box Response Time and Accuracy

This overall score is different from the performance plot, which shows only the percentage of area covered.

Player with the high score wins a $100 Amazon Gift Card!

SEARCH, TRACK, DESTROY

46/47

Interactive Practice

You will now have approx. **15 minutes to practice** with the actual interface and to ask questions.

After practicing, you will perform as the multi-UxV operator for 2 **twenty-minute missions**.

HAVE FUN!

47/47

300

Appendix R: Proficiency Tests

Appendix S: Questionnaires

Scenario Feedback Survey

Round 1

1. Subject number:_____

2. How confident were you about your performance?

   Not Confident   Somewhat Confident   Confident   Very Confident   Extremely Confident

   Comments:

3. How busy did you feel during the mission?

   Idle   Not Busy   Busy   Very Busy   Extremely Busy

4. How satisfied were you with the plans created by the Automated Scheduler?

   Very Unsatisfied   Unsatisfied   Satisfied   Very satisfied   Extremely satisfied

Scenario Feedback Survey

Round 2

1. Subject number:_____

2. How confident were you about your performance?

   Not Confident   *Somewhat Confident*   *Confident*   *Very Confident*   *Extremely Confident*

   Comments:

3. How busy did you feel during the mission?

   Idle   *Not Busy*   *Busy*   *Very Busy*   *Extremely Busy*

4. How satisfied were you with the plans created by the Automated Scheduler?

   Very Unsatisfied   *Unsatisfied*   *Satisfied*   *Very satisfied*   *Extremely satisfied*

Questions about the Experiment Overall

1. Were there aspects of the interface that you particularly liked or disliked?

2. Did you feel that the Automated Scheduler was fast enough? Did it make good plans?

3. Were there times that your trust in the Automated Scheduler increased or decreased? Why?

4. Did you look at the performance plot often? How did it make you feel about how well you were doing? Did you change your behavior based on what you saw there?

5. Other comments:

Scenario Feedback Survey

Round 2

Subject number:_____

	Not at all						Extremely
1. The automated scheduler is deceptive	1	2	3	4	5	6	7
2. The automated scheduler behaves in an underhanded manner	1	2	3	4	5	6	7
3. I am suspicious of the automated scheduler's intent, action, or outputs	1	2	3	4	5	6	7
4. I am wary of the automated scheduler	1	2	3	4	5	6	7
5. The automated scheduler's actions will have a harmful of injurious outcome	1	2	3	4	5	6	7
6. I am confident in the automated scheduler	1	2	3	4	5	6	7
7. The automated scheduler provides security	1	2	3	4	5	6	7
8. The automated scheduler has integrity	1	2	3	4	5	6	7
9. The automated scheduler is dependable	1	2	3	4	5	6	7
10. The automated scheduler is reliable	1	2	3	4	5	6	7
11. I can trust the automated scheduler	1	2	3	4	5	6	7
12. I am familiar with the automated scheduler	1	2	3	4	5	6	7

Appendix T: Detailed Experiment Statistical Analysis and Descriptive Statistics

This appendix presents the statistical results of the experiment described in Chapter 5. The experiment included three independent variables: *A Priori* Priming Level (Positive Priming, Negative Priming, No Priming), Real-time Priming Level (Low or High), and Information Time Delay (No Delay or With Delay). Numerous dependent variables were considered in the analysis of the data in order to capture and measure performance, workload, SA, and subjective ratings of performance, workload, confidence, and trust, as described in Chapter 5. An analysis of the dependent variables and all descriptive statistics is presented below.

Statistical Analysis Overview

All dependent variables were recorded by the computer simulation. For Area Coverage and Percentage of Time that Targets were Tracked, a 3 x 2 x 2 repeated measures Analysis of Variance (ANOVA) model was used ($\alpha = 0.05$). These parametric dependent variables met the homogeneity of variance and normality assumptions of the ANOVA model. For three measures of primary workload (Utilization, Total Mouse Clicks, Length of Time to Replan), a MANOVA model was used because these dependent variables were moderated correlated. All other dependent variables, including reaction times and Likert scale data, did not meet ANOVA assumptions, and non-parametric analyses were used. All significant tests are underlined.

Order Effects

First, it should be noted that Information Time Delay level was a within-subjects variable, so every operator experienced both Information Time Delay levels with the order of presentation counterbalanced and randomized. All dependent variables were tested for order effects and there were only two dependent variables with significant order effects. MANOVA results showed that operators had a 16% lower average length of time to replan in the second mission as compared to the first mission, $F(1,71) = 4.659$, $p = 0.034$. Also, a Mann Whitney comparison showed that operators had a 21% faster reaction time to all embedded secondary tasks in the second mission as compared to the first mission ($Z = -2.169$, $p = 0.030$). Operators learned to replan faster between the two trials and had faster reaction times to embedded secondary tasks in the second

trial. While notable, these learning effects for using the interface did not have a significant impact on overall area system performance or other operator actions or ratings (Table 27), thus the analysis proceeded without including an order effect factor in ANOVA and MANOVA models, to increase the degrees of freedom in these analyses.

Table 27. Summary of statistical tests for order effects.

Metric	Statistical Test	Main Effect for Order (2 levels: First and Second Order)
Area Coverage	Repeated Measures ANOVA	$F(1,35) = 0.319$, $p = 0.576$
Targets Found	Mann Whitney	$Z = -1.892$, $p = 0.058$
% Time Targets Tracked	Repeated Measures ANOVA	$F(1,35) = 0.311$, $p = 0.581$
Correct Hostiles Destroyed	Mann Whitney	$Z = -1.809$, $p = 0.070$
Mistaken Hostiles Destroyed	Mann Whitney	$Z = -1.721$, $p = 0.085$
Utilization	MANOVA	$F(1,71) = 2.421$, $p = 0.124$
Click Count	MANOVA	$F(1,71) = 0.226$, $p = 0.636$
Average Length of Time to Replan	MANOVA	$\underline{F(1,71) = 4.659, p = 0.034}$
Chat Question Accuracy	Mann Whitney	$Z = -1.308$, $p = 0.191$
Target Re-designation Accuracy	Mann Whitney	$Z = -0.112$, $p = 0.911$
Embedded Secondary Tasks	Mann Whitney	$\underline{Z = -2.169, p = 0.030}$
Average Performance Rating	Mann Whitney	$Z = -0.250$, $p = 0.803$
Average Expectations Rating	Mann Whitney	$Z = -0.205$, $p = 0.838$
Average PPG Rating	Mann Whitney	$Z = -0.970$, $p = 0.332$
Average Trust Rating	Mann Whitney	$Z = -0.288$, $p = 0.774$
Confidence Rating	Mann Whitney	$Z = -1.587$, $p = 0.113$
Workload Rating	Mann Whitney	$Z = -0.707$, $p = 0.480$
Satisfaction with AS plans	Mann Whitney	$Z = -0.510$, $p = 0.610$

Mission Performance

Mission performance was measured by overall mission performance metrics, computed at the end of the mission as well as errors made by operators who either destroyed friendly targets or who destroyed hostile targets against the Rules of Engagement. There were no significant correlations between these dependent variables. Additionally, Metacognitive Awareness Inventory (MAI) Score, a demographic variable, correlated with Area Coverage ($\rho = -0.285$, $p = 0.005$) and Percentage of Time that Targets were Tracked ($\rho = -0.201$, $p = 0.050$). While these relationships were weak to moderate by human factors standards, MAI Score was used as a covariate in the ANOVA models for these two dependent variables. MAI Score was included in model to reduce error variance, not to investigate MAI Score as a primary research question.

Finally, Targets Found, Correct Hostiles Destroyed, and Mistaken Hostiles Destroyed were evaluated with non-parametric tests.

- Area Coverage:
  - *A Priori* Priming Level, $F(2,41) = 0.016$, $p = 0.984$
  - Real-time Priming Level, $F(1,41) = 0.002$, $p = 0.961$
  - Information Time Delay, $F(1,41) = 0.495$, $p = 0.486$
  - <u>MAI Score, $F(1,41) = 5.885$, $p = 0.020$</u>
  - *A Priori* Priming Level*Real-time Priming Level, $F(2,41) = 1.305$, $p = 0.282$
  - *A Priori* Priming Level*Information Time Delay, $F(2,41) = 2.219$, $p = 0.122$
  - Real-time Priming Level*Information Time Delay, $F(1,41) = 2.673$, $p = 0.110$
  - Information Time Delay*MAI Score, $F(1,41) = 0.930$, $p = 0.341$
  - *A Priori* Priming Level*Real-time Priming Level*Information Time Delay, $F(2,41) = 1.396$, $p = 0.259$
- Targets Found
  - *A Priori* Priming Level, $\chi^2(2, N=96) = 0.127$, $p = 0.939$ (Kruskal-Wallis omnibus)
  - Real-time Priming Level, $Z = -0.157$, $p = 0.875$, (Mann-Whitney Independent)
  - Information Time Delay, $Z = -1.293$, $p = 0.196$, (Mann-Whitney Dependent)
- Percentage of Time that Targets were Tracked
  - *A Priori* Priming Level, $F(2,41) = 0.501$, $p = 0.610$
  - Real-time Priming Level, $F(1,41) = 3.403$, $p = 0.072$
  - Information Time Delay, $F(1,41) = 0.002$, $p = 0.968$
  - <u>MAI Score, $F(1,41) = 5.105$, $p = 0.029$</u>
  - *A Priori* Priming Level*Real-time Priming Level, $F(2,41) = 0.233$, $p = 0.793$
  - *A Priori* Priming Level*Information Time Delay, $F(2,41) = 1.058$, $p = 0.356$
  - Real-time Priming Level*Information Time Delay, $F(1,41) = 0.603$, $p = 0.442$
  - Information Time Delay*MAI Score, $F(1,41) = 0.118$, $p = 0.773$
  - <u>*A Priori* Priming Level*Real-time Priming Level*Information Time Delay, $F(2,41) = 3.992$, $p = 0.026$</u>
- Correct Hostiles Destroyed
  - *A Priori* Priming Level, $\chi^2(2, N=96) = 2.676$, $p = 0.262$ (Kruskal-Wallis omnibus)
  - Real-time Priming Level, $Z = -0.883$, $p = 0.377$, (Mann-Whitney Independent)
  - Information Time Delay, $Z = -0.860$, $p = 0.390$, (Mann-Whitney Dependent)

- Mistaken Hostiles Destroyed
  - *A Priori* Priming Level, $\chi^2(2, N=96) = 14.611$, p = 0.001 (Kruskal-Wallis omnibus)
    - Mann-Whitney pairwise comparisons:
    - Negative Priming-No Priming: Z = -2.747, p = 0.006
    - Negative Priming-Positive Priming: Z = -3.101, p = 0.002
    - No Priming-Positive Priming: Z = -0.568, p = 0.570
  - Real-time Priming Level, Z = -0.209, p = 0.834, (Mann-Whitney Independent)
  - Information Time Delay, Z = -0.324, p = 0.746, (Mann-Whitney Dependent)

Table 28. Performance Metrics Summary for *A Priori* Priming

Metric	*A Priori* Priming	Mean	Median	Std Dev	Min	Max
% Area Coverage	Negative	62.9%	61.9%	8.6%	46.8%	80.0%
	No Priming	63.7%	63.3%	7.1%	43.4%	78.8%
	Positive	62.4%	62.6%	9.9%	39.6%	85.0%
% Targets Found	Negative	86.6%	90.0%	9.7%	60.0%	100%
	No Priming	85.3%	90.0%	11.1%	60.0%	100%
	Positive	85.9%	90.0%	11.6%	60.0%	100%
% Time Targets Tracked	Negative	88.7%	88.9%	7.0%	66.5%	100%
	No Priming	87.5%	88.2%	6.2%	73.9%	97.1%
	Positive	87.6%	88.9%	7.2%	68.7%	96.7%
Correct Hostiles Destroyed	Negative	2.9	3.0	1.1	0.0	5.0
	No Priming	3.1	3.0	1.2	1.0	5.0
	Positive	3.4	3.0	0.8	2.0	5.0
Mistaken Hostiles Destroyed	Negative	0.5	0.0	0.8	0.0	2.0
	No Priming	0.1	0.0	0.4	0.0	2.0
	Positive	0.1	0.0	0.4	0.0	2.0

Table 29. Performance Metrics Summary for Real-Time Priming

Metric	Real-Time Priming	Mean	Median	Std Dev	Min	Max
% Area Coverage	Low	62.8%	64.3%	9.2%	39.6%	80.0%
	High	63.2%	61.4%	7.9%	48.9%	85.0%
% Targets Found	Low	85.8%	90.0%	10.5%	60.0%	100%
	High	86.0%	90.0%	11.1%	60.0%	100%
% Time Targets Tracked	Low	89.0%	90.6%	6.5%	73.9%	100%
	High	86.8%	88.2%	7.0%	66.5%	97.7%
Correct Hostiles Destroyed	Low	3.3	3.0	1.0	1.0	5.0
	High	3.0	3.0	1.1	0.0	5.0
Mistaken Hostiles Destroyed	Low	0.2	0.0	0.4	0.0	2.0
	High	0.3	0.0	0.7	0.0	2.0

Table 30. Performance Metrics Summary for Information Time Delay

Metric	Info Time Delay	Mean	Median	Std Dev	Min	Max
% Area Coverage	No Delay	60.8%	60.5%	9.3%	39.6%	85.0%
	With Delay	65.1%	64.3%	7.2%	43.4%	83.4%
% Targets Found	No Delay	87.3%	90.0%	10.9%	60.0%	100%
	With Delay	84.6%	90.0%	10.5%	60.0%	100%
% Time Targets Tracked	No Delay	90.0%	91.2%	6.1%	74.5%	100%
	With Delay	85.8%	86.9%	6.8%	66.5%	97.7%
Correct Hostiles Destroyed	No Delay	3.1	3.0	1.1	1.0	5.0
	With Delay	3.2	3.0	1.0	0.0	5.0
Mistaken Hostiles Destroyed	No Delay	0.2	0.0	0.6	0.0	2,0
	With Delay	0.2	0.0	0.5	0.0	2.0

Workload

Primary workload was measured through utilization, calculating the ratio of the total operator "busy time" to total mission time. Average Length of Time to Replan was evaluated as a component of workload. After the experiment, an evaluation of the total number of mouse clicks by each operator during the mission was conducted to further analyze workload. This measure

was not part of the original experiment design and its use is for post-hoc analysis only. As these three variables were moderated correlated, a MANOVA model was used for analyzing these three dependent variables. Additionally, Metacognitive Awareness Inventory (MAI) Score, a demographic variable, correlated with Average Length of Time to Replan (ρ=0.229, p = 0.025). While this relationship was weak to moderate by human factors standards, MAI Score was used as a covariate in the MANOVA model. MAI Score was included in model to reduce error variance, not to investigate MAI Score as a primary research question

In addition to these primary workload metrics, secondary workload was measured via reaction times to chat message information queries, as well as reaction times when instructed to create search tasks via the chat tool. As the chat messages and prompted search tasks were different and appeared at different times for the two missions performed by each operator, no statistical analysis was performed on the within-subjects variable of Information Time Delay for the reaction times. These reaction times did not satisfy the ANOVA assumptions, so non-parametric tests were used.

- Utilization
  - *A Priori* Priming Level, $F(2,83) = 1.444$, p = 0.242
  - Real-time Priming Level, $F(1,83) = 0.000$, p = 0.986
  - Information Time Delay, $F(1,83) = 0.005$, p = 0.946
  - MAI Score, $F(1,83) = 0.609$, p = 0.437
  - *A Priori* Priming Level*Real-time Priming Level, $F(2,83) = 0.663$, p = 0.518
  - *A Priori* Priming Level*Information Time Delay, $F(2,83) = 0.399$, p = 0.672
  - Real-time Priming Level*Information Time Delay, $F(1,83) = 0.351$, p = 0.555
  - *A Priori* Priming Level*Real-time Priming Level*Information Time Delay, $F(2,83) = 0.956$, p = 0.389
- Total Mouse Clicks
  - *A Priori* Priming Level, $F(2,83) = 2.089$, p = 0.130
  - Real-time Priming Level, $F(1,83) = 0.799$, p = 0.374
  - Information Time Delay, $F(1,83) = 0.158$, p = 0.692
  - MAI Score, $F(1,83) = 0.255$, p = 0.615
  - *A Priori* Priming Level*Real-time Priming Level, $F(2,83) = 1.313$, p = 0.275
  - *A Priori* Priming Level*Information Time Delay, $F(2,83) = 0.910$, p = 0.406

- o Real-time Priming Level*Information Time Delay, F(1,83) = 0.001, p = 0.972
- o *A Priori* Priming Level*Real-time Priming Level*Information Time Delay, F(2,83) = 0.635, p = 0.532
- Average Length of Time to Replan
  - o *A Priori* Priming Level, F(2,83) = 0.498, p = 0.610
  - o Real-time Priming Level, F(1,83) = 0.029, p = 0.865
  - o Information Time Delay, F(1,83) = 0.000, p = 0.989
  - o <u>MAI Score, F(1,83) = 4.241, p = 0.043</u>
  - o *A Priori* Priming Level*Real-time Priming Level, F(2,83) = 0.810, p = 0.448
  - o *A Priori* Priming Level*Information Time Delay, F(2,83) = 0.386, p = 0.681
  - o Real-time Priming Level*Information Time Delay, F(1,83) = 0.440, p = 0.509
  - o *A Priori* Priming Level*Real-time Priming Level*Information Time Delay, F(2,83) = 0.432, p = 0.651
- Chat #1 reaction
  - o <u>*A Priori* Priming Level, χ^2(2, N=96) = 10.065, p = 0.007 (Kruskal-Wallis omnibus)</u>
    - ▪ Mann-Whitney pairwise comparisons:
    - ▪ <u>Negative Priming-No Priming: Z = -2.135, p = 0.033</u>
    - ▪ <u>Negative Priming-Positive Priming: Z = -3.115, p = 0.002</u>
    - ▪ No Priming-Positive Priming: Z = -0.873, p = 0.383
  - o Real-time Priming Level, Z = -0.799, p = 0.424, (Mann-Whitney Independent)
- Chat #2 reaction
  - o <u>*A Priori* Priming Level, χ^2(2, N=96) = 7.491, p = 0.024 (Kruskal-Wallis omnibus)</u>
    - ▪ Mann-Whitney pairwise comparisons:
    - ▪ Negative Priming-No Priming: Z = -0.054, p = 0.957
    - ▪ <u>Negative Priming-Positive Priming: Z = -2.391, p = 0.017</u>
    - ▪ <u>No Priming-Positive Priming: Z = -2.337, p = 0.019</u>
  - o Real-time Priming Level, Z = -1.301, p = 0.193, (Mann-Whitney Independent)
- Chat #3 reaction
  - o *A Priori* Priming Level, χ^2(2, N=96) = 1.608, p = 0.447 (Kruskal-Wallis omnibus)
  - o <u>Real-time Priming Level, Z = -3.019, p = 0.003, (Mann-Whitney Independent)</u>
- Search task #1 reaction
  - o *A Priori* Priming Level, χ^2(2, N=96) = 2.514, p = 0.284 (Kruskal-Wallis omnibus)
  - o Real-time Priming Level, Z = -1.636, p = 0.102, (Mann-Whitney Independent)
- Search task #2 reaction

- o *A Priori* Priming Level, $\chi^2(2, N=96) = 5.469$, p = 0.065 (Kruskal-Wallis omnibus)
    - ▪ Mann-Whitney pairwise comparisons:
    - ▪ Negative Priming-No Priming: Z = -0.762, p = 0.446
    - ▪ <u>Negative Priming-Positive Priming: Z = -2.297, p = 0.022</u>
    - ▪ No Priming-Positive Priming: Z = -1.513, p = 0.130
- o <u>Real-time Priming Level, Z = -2.735, p = 0.006, (Mann-Whitney Independent)</u>

Table 31. Workload Metrics Summary for *A Priori* Priming

Metric	*A Priori* Priming	Mean	Median	Std Dev	Min	Max
Utilization	Negative	47.8%	47.9%	5.30%	36.6%	61.0%
	No Priming	46.7%	46.2%	8.90%	30.6%	65.7%
	Positive	44.8%	45.3%	8.00%	31.0%	61.6%
Total Mouse Clicks	Negative	359	356	75.2	233	527
	No Priming	338	327	80.6	206	532
	Positive	318	296	84.8	186	542
Length of Time to Replan	Negative	7.40	6.90	2.70	3.36	12.4
	No Priming	7.50	6.60	3.60	2.15	15.5
	Positive	7.20	7.10	2.90	2.00	13.2
Chat #1 Reaction Time	Negative	19.8	15.7	12.7	4.43	51.1
	No Priming	13.9	11.1	8.90	4.98	38.7
	Positive	12.3	9.40	8.80	4.81	43.3
Chat #2 Reaction Time	Negative	26.9	21.3	16.0	7.30	60.0
	No Priming	27.4	24.2	16.7	7.55	60.0
	Positive	18.4	15.7	10.5	7.19	60.0
Chat #3 Reaction Time	Negative	13.6	10.5	8.00	4.78	42.5
	No Priming	18.2	12.6	15.7	4.80	60.0
	Positive	14.4	9.60	13.5	4.91	60.0
Prompted Search Task #1 Reaction Time	Negative	27.8	20.0	21.4	3.14	60.0
	No Priming	27.6	17.3	20.5	5.41	60.0
	Positive	35.0	23.8	22.6	8.26	60.0
Prompted Search Task #2 Reaction Time	Negative	33.0	26.5	22.1	7.77	60.0
	No Priming	28.8	18.1	21.2	6.69	60.0
	Positive	18.2	13.7	13.0	6.44	60.0

Table 32. Workload Metrics Summary for Real-Time Priming

Metric	Real-Time Priming	Mean	Median	Std Dev	Min	Max
Utilization	Low	46.5%	47.6%	7.60%	30.6%	65.7%
	High	46.4%	46.3%	7.60%	31.0%	63.5%
Total Mouse Clicks	Low	331	330	79.4	206	542
	High	345	331	83.1	186	532
Length of Time to Replan	Low	7.50	7.00	3.10	3.18	15.5
	High	7.20	6.90	3.10	2.00	13.2
Chat #1 Reaction Time	Low	14.9	10.1	10.8	4.44	50.9
	High	15.7	11.1	10.7	5.50	51.1
Chat #2 Reaction Time	Low	26.9	21.2	16.9	7.19	60.0
	High	21.6	18.9	12.7	7.55	60.0
Chat #3 Reaction Time	Low	18.7	14.0	14.9	4.78	60.0
	High	12.1	8.90	9.50	5.23	57.5
Prompted Search Task #1 Reaction Time	Low	33.7	23.1	23.0	3.14	60.0
	High	26.7	18.0	19.6	5.41	60.0
Prompted Search Task #2 Reaction Time	Low	32.7	20.1	21.8	6.44	60.0
	High	20.6	13.1	16.1	6.69	60.0

Table 33. Workload Metrics Summary for Information Time Delay

Metric	Info Time Delay	Mean	Median	Std Dev	Min	Max
Utilization	No Delay	46.5%	47.2%	7.00%	30.6%	61.6%
	With Delay	46.4%	46.1%	8.20%	31.0%	65.7%
Total Mouse Clicks	No Delay	342	336	83.7	206	542
	With Delay	335	327	79.2	186	532
Length of Time to Replan	No Delay	7.40	6.70	2.90	2.00	13.2
	With Delay	7.40	7.10	3.30	2.15	15.5

Situation Awareness

Situation Awareness (SA) was measured through two metrics: the accuracy of responses to periodic chat box messages querying the participant about aspects of the mission and the

accuracy of re-designations of unknown targets based on chat intelligence information. For both metrics, non-parametric tests were needed.

- Chat Question Accuracy
  - *A Priori* Priming Level, χ^2(2, N=96) = 3.308, p = 0.191 (Kruskal-Wallis omnibus)
  - Real-time Priming Level, Z = -1.795, p = 0.073, (Mann-Whitney Independent)
  - Information Time Delay, Z = -0.022, p = 0.982, (Mann-Whitney Dependent)
- Target ID Re-Designation Accuracy
  - *A Priori* Priming Level, χ^2(2, N=96) = 0.498, p = 0.779 (Kruskal-Wallis omnibus)
  - Real-time Priming Level, Z = -0.965, p = 0.335, (Mann-Whitney Independent)
  - <u>Information Time Delay, Z = -3.009, p = 0.003, (Mann-Whitney Dependent)</u>

Table 34. SA Metrics Summary for *A Priori* Priming

Metric	*A Priori* Priming	Mean	Median	Std Dev	Min	Max
Chat Question Accuracy	Negative	58.5%	67.0%	22.6%	0.00%	100%
	No Priming	65.8%	67.0%	27.5%	0.00%	100%
	Positive	68.9%	67.0%	18.9%	33.0%	100%
Target Re-Designation Accuracy	Negative	60.9%	66.7%	26.3%	0.00%	100%
	No Priming	59.9%	58.4%	31.6%	0.00%	100%
	Positive	65.1%	66.7%	27.8%	0.00%	100%

Table 35. SA Metrics Summary for Real-Time Priming

Metric	Real-Time Priming	Mean	Median	Std Dev	Min	Max
Chat Question Accuracy	Low	68.3%	67.0%	22.8%	0.00%	100%
	High	60.5%	67.0%	23.6%	0.00%	100%
Target Re-Designation Accuracy	Low	58.7%	66.7%	29.2%	0.00%	100%
	High	65.3%	66.7%	27.5%	25.0%	100%

Table 36. SA Metrics Summary for Information Time Delay

Metric	Info Time Delay	Mean	Median	Std Dev	Min	Max
Chat Question Accuracy	No Delay	64.0%	67.0%	21.7%	0.00%	100%
	With Delay	64.7%	67.0%	25.2%	0.00%	100%
Target Re-Designation Accuracy	No Delay	71.4%	75.0%	29.8%	0.00%	100%
	With Delay	52.6%	50.0%	23.8%	0.00%	100%

Real-time Subjective Responses

Throughout the mission, a pop-up survey window appeared in the lower left corner of the Map Display to ask the operator to provide these three ratings:

- How well you think the system is performing (1-7)
- How well you expect the system to perform (1-7)
- Your trust in the Automated Scheduler (1-7)

A fourth metric, the Perceived Performance Gap (PPG) was calculated by taking the percent difference between the expectation and performance rating. For the analysis below, all ratings taken during the mission were averaged. Non-parametric tests were needed for this Likert scale data.

- Performance
  - <u>*A Priori* Priming Level, χ^2(2, N=94) = 5.774, p = 0.056 (Kruskal-Wallis omnibus)</u>
    - Mann-Whitney pairwise comparisons:
    - <u>Negative Priming-No Priming: Z = -1.832, p = 0.067</u>
    - <u>Negative Priming-Positive Priming: Z = -2.160, p = 0.031</u>
    - No Priming-Positive Priming: Z = -0.833, p = 0.405
  - <u>Real-time Priming Level, Z = -2.122, p = 0.034, (Mann-Whitney Independent)</u>
  - Information Time Delay, Z = -0.213, p = 0.831, (Mann-Whitney Dependent)
- Expectations
  - <u>*A Priori* Priming Level, χ^2(2, N=94) = 11.767, p = 0.003 (Kruskal-Wallis omnibus)</u>
    - Mann-Whitney pairwise comparisons:
    - Negative Priming-No Priming: Z = -0.395, p = 0.693
    - <u>Negative Priming-Positive Priming: Z = -3.189, p = 0.001</u>

- No Priming-Positive Priming: $Z = -2.674$, p = 0.007
  - Real-time Priming Level, $Z = -2.145$, p = 0.032, (Mann-Whitney Independent)
  - Information Time Delay, $Z = -1.204$, p = 0.229, (Mann-Whitney Dependent)
- PPG
  - *A Priori* Priming Level, $\chi^2(2, N=94) = 4.520$, p = 0.104 (Kruskal-Wallis omnibus)
  - Real-time Priming Level, $Z = -0.314$, p = 0.753, (Mann-Whitney Independent)
  - Information Time Delay, $Z = -0.225$, p = 0.822, (Mann-Whitney Dependent)
- Trust
  - *A Priori* Priming Level, $\chi^2(2, N=94) = 14.740$, p = 0.001 (Kruskal-Wallis omnibus)
    - Mann-Whitney pairwise comparisons:
    - Negative Priming-No Priming: $Z = -1.036$, p = 0.300
    - Negative Priming-Positive Priming: $Z = -3.741$, p < 0.001
    - No Priming-Positive Priming: $Z = -2.614$, p = 0.009
  - Real-time Priming Level, $Z = -0.851$, p = 0.395, (Mann-Whitney Independent)
  - Information Time Delay, $Z = -0.895$, p = 0.371, (Mann-Whitney Dependent)

Table 37. Real-time Survey Metrics Summary for *A Priori* Priming

Metric	*A Priori* Priming	Mean	Median	Std Dev	Min	Max
Performance	Negative	4.80	4.70	0.88	3.20	7.00
	No Priming	5.16	5.25	0.93	3.00	6.90
	Positive	5.33	5.50	0.98	3.40	6.70
Expectations	Negative	5.27	5.20	0.87	3.44	6.50
	No Priming	5.40	5.30	0.87	4.00	7.00
	Positive	5.96	6.00	0.66	4.73	7.00
PPG	Negative	7.16%	8.00%	13.1%	-16.0%	35.0%
	No Priming	3.65%	2.90%	14.6%	-23.0%	42.0%
	Positive	10.5%	7.70%	11.5%	-2.00%	35.0%
Trust	Negative	4.46	4.70	1.01	2.20	6.00
	No Priming	4.82	4.80	1.04	2.60	6.90
	Positive	5.54	5.80	1.17	2.80	7.00

Table 38. Real-time Survey Metrics Summary for Real-Time Priming

Metric	Real-Time Priming	Mean	Median	Std Dev	Min	Max
Performance	Low	5.31	5.30	0.95	3.00	7.00
	High	4.88	4.90	0.89	3.27	6.50
Expectations	Low	5.73	6.00	0.85	4.10	7.00
	High	5.36	5.20	0.82	3.44	7.00
PPG	Low	6.70%	5.00%	12.04%	-15.0%	36.0%
	High	7.60%	4.00%	14.50%	-23.0%	42.0%
Trust	Low	5.08	4.90	1.14	2.60	7.00
	High	4.81	4.90	1.17	2.20	6.80

Table 39. Real-time Survey Metrics Summary for Information Time Delay

Metric	Info Time Delay	Mean	Median	Std Dev	Min	Max
Performance	No Delay	5.11	5.00	0.95	3.27	6.90
	With Delay	5.09	5.20	0.95	3.00	7.00
Expectations	No Delay	5.58	5.64	0.83	3.50	7.00
	With Delay	5.52	5.83	0.88	3.44	7.00
PPG	No Delay	7.55%	4.00%	12.38%	-15.0%	42.0%
	With Delay	6.76%	5.00%	14.21%	-23.0%	36.0%
Trust	No Delay	4.99	4.90	1.13	2.20	7.00
	With Delay	4.91	4.90	1.20	2.60	7.00

Pre- and Post-Mission Subjective Responses

First, a survey immediately following the practice mission asked operators to rate their trust in the AS on a scale from 1-7 (low to high). Next, at the end of each mission, a survey was provided asking the participant for a subjective rating of his or her confidence, workload, and satisfaction with the plans generated by the AS on a Likert scale from 1-5 (1 low, 5 high). Finally, at the end of the entire experiment, all test subjects filled out a 12-question survey which is commonly used to measure trust in automation and has been empirically validated (Jian, et al., 2000). Since each test subject experienced both Information Delay Levels, there is no analysis of the differences in the pre-experiment or post-experiment trust survey results between Information Delay Levels. Non-parametric tests were needed for this Likert scale data.

- Pre-Experiment Trust
  - *A Priori* Priming Level, $\chi^2(2, N=47) = 11.636$, p = 0.003 (Kruskal-Wallis omnibus)
    - Mann-Whitney pairwise comparisons:
    - Negative Priming-No Priming: Z = -0.807, p = 0.420
    - Negative Priming-Positive Priming: Z = -3.186, p = 0.002
    - No Priming-Positive Priming: Z = -2.570, p = 0.010
  - Real-time Priming Level, Z = -0.993, p = 0.321, (Mann-Whitney Independent)
- Confidence
  - *A Priori* Priming Level, $\chi^2(2, N=96) = 0.108$, p = 0.947 (Kruskal-Wallis omnibus)
  - Real-time Priming Level, Z = -4.462, p < 0.001, (Mann-Whitney Independent)
  - Information Time Delay, Z = -1.483, p = 0.138, (Mann-Whitney Dependent)
- Workload
  - *A Priori* Priming Level, $\chi^2(2, N=96) = 1.116$, p = 0.572 (Kruskal-Wallis omnibus)
  - Real-time Priming Level, Z = -0.707, p = 0.480, (Mann-Whitney Independent)
  - Information Time Delay, Z = -0.354, p = 0.723, (Mann-Whitney Dependent)
- Satisfaction with AS Plans
  - *A Priori* Priming Level, $\chi^2(2, N=96) = 1.073$, p = 0.585 (Kruskal-Wallis omnibus)
  - Real-time Priming Level, Z = -0.589, p = 0.556, (Mann-Whitney Independent)
  - Information Time Delay, Z = -0.592, p = 0.554, (Mann-Whitney Dependent)
- Post-Experiment Trust Survey
  - *A Priori* Priming Level, $\chi^2(2, N=48) = 1.986$, p = 0.371 (Kruskal-Wallis omnibus)
  - Real-time Priming Level, Z = -0.722, p = 0.470, (Mann-Whitney Independent)

Table 40. Post-Mission Survey Metrics Summary for *A Priori* Priming

Metric	*A Priori* Priming	Mean	Median	Std Dev	Min	Max
Pre-Experiment Trust Survey	Negative	5.13	5.00	0.92	3.00	6.00
	No Priming	5.44	5.50	0.81	4.00	7.00
	Positive	6.19	6.00	0.66	5.00	7.00
Confidence	Negative	2.50	2.00	0.98	1.00	5.00
	No Priming	2.47	2.00	0.98	1.00	5.00
	Positive	2.38	2.00	0.91	1.00	4.00
Workload	Negative	2.97	3.00	0.60	2.00	4.00
	No Priming	3.06	3.00	0.72	2.00	4.00
	Positive	2.84	3.00	0.77	1.00	4.00
Satisfaction with AS Plans	Negative	2.63	3.00	0.79	1.00	4.00
	No Priming	2.91	3.00	1.00	1.00	5.00
	Positive	2.69	3.00	0.69	1.00	4.00
Post-Experiment Trust Survey	Negative	9.27	9.00	12.9	-16.0	28.0
	No Priming	14.1	15.0	13.1	-20.0	31.0
	Positive	15.3	16.5	15.1	-20.0	36.0

Table 41. Post-Mission Survey Metrics Summary for Real-Time Priming

Metric	Real-Time Priming	Mean	Median	Std Dev	Min	Max
Pre-Experiment Trust Survey	Low	5.46	6.00	0.93	3.00	7.00
	High	5.74	6.00	0.86	4.00	7.00
Confidence	Low	2.88	3.00	0.89	1.00	5.00
	High	2.02	3.00	0.81	1.00	4.00
Workload	Low	3.02	3.00	0.73	2.00	4.00
	High	2.90	3.00	0.66	1.00	4.00
Satisfaction with AS Plans	Low	2.79	3.00	0.82	1.00	4.00
	High	2.69	3.00	0.85	1.00	5.00
Post-Experiment Trust Survey	Low	13.8	14.0	15.9	-16.0	30.0
	High	12.1	13.0	11.3	-20.0	36.0

Table 42. Post-Mission Survey Metrics Summary for Information Time Delay

Metric	Info Time Delay	Mean	Median	Std Dev	Min	Max
Confidence	No Delay	2.38	2.00	0.92	1.00	5.00
	With Delay	2.57	2.00	0.95	1.00	5.00
Workload	No Delay	2.94	3.00	0.76	1.00	4.00
	With Delay	2.96	3.00	0.62	2.00	4.00
Satisfaction with AS Plans	No Delay	2.70	3.00	0.86	1.00	5.00
	With Delay	2.77	3.00	0.84	1.00	5.00

Gamer vs. Nongamer Click Count Analysis

Descriptive statistics for comparison of click count between gamers and nongamers across the different A Priori Priming levels are shown in Table 43.

Table 43. Descriptive statistics for gamers vs. nongamer click count analysis.

Gaming Type	*A Priori* Priming	Mean	Median	Std Dev	Min	Max
Gamers	Negative	381	366	95.6	251	527
	No Priming	299	316	62.7	206	407
	Positive	277	269	42.1	229	347
Nongamers	Negative	348	354	63.8	233	467
	No Priming	368	356	81.5	217	532
	Positive	343	335	94.6	186	542

Statistical Analysis Summary

The results of all omnibus factor level tests are summarized in Table 44, where the conditions with statistically significant results are shown in bold.

Table 44. Summary of Experimental Findings

Category	Metric	*A Priori* Priming	Real-Timing Priming	Information Delay
System Performance	% Area Coverage	Indistinguishable (p = 0.984)	Indistinguishable (p = 0.961)	Indistinguishable (p = 0.486)
	% Targets Found	Indistinguishable (p = 0.939)	Indistinguishable (p = 0.875)	Indistinguishable (p = 0.196)
	% Time Targets Tracked	Indistinguishable (p = 0.610)	Indistinguishable (p = 0.072)	Indistinguishable (p = 0.968)
	Correct Hostiles Destroyed	Indistinguishable (p = 0.262)	Indistinguishable (p = 0.377)	Indistinguishable (p = 0.390)
	Mistaken Hostiles Destroyed	**Significant difference (p = 0.001)**	Indistinguishable (p = 0.834)	Indistinguishable (p = 0.746)
Primary Workload	Utilization	Indistinguishable (p = 0.242)	Indistinguishable (p = 0.986)	Indistinguishable (p = 0.946)
	Total Click Count	Indistinguishable (p = 0.130)	Indistinguishable (p = 0.374)	Indistinguishable (p = 0.692)
	Average Length of Time to Replan	Indistinguishable (p = 0.610)	Indistinguishable (p = 0.865)	Indistinguishable (p = 0.989)
Secondary Workload	Chat #1 reaction time	**Significant difference (p = 0.007)**	Indistinguishable (p = 0.424)	N/A
	Chat #2 reaction time	**Significant difference (p = 0.024)**	Indistinguishable (p = 0.193)	N/A
	Chat #3 reaction time	Indistinguishable (p = 0.447)	**Significant difference (p = 0.003)**	N/A
	Prompted Search Task #1 reaction time	Indistinguishable (p = 0.284)	Indistinguishable (p = 0.102)	N/A
	Prompted Search Task #2 reaction time	Positive Priming (p = 0.065)	**Significant difference (p = 0.006)**	N/A
Situation Awareness	Chat question accuracy	Indistinguishable (p = 0.191)	Low (p = 0.073)	Indistinguishable (p = 0.982)
	Target re-designation accuracy	Indistinguishable (p = 0.779)	Indistinguishable (p = 0.335)	**Significant difference (p = 0.003)**
Real-Time Subjective Ratings	Average Performance	**Significant difference (p = 0.056)**	**Significant difference (p = 0.034)**	Indistinguishable (p = 0.831)
	Average Expectations	**Significant difference (p = 0.003)**	**Significant difference (p = 0.032)**	Indistinguishable (p = 0.229)
	Average PPG	Indistinguishable (p = 0.104)	Indistinguishable (p = 0.753)	Indistinguishable (p = 0.822)
	Average Trust	**Significant difference (p = 0.001)**	Indistinguishable (p = 0.395)	Indistinguishable (p = 0.371)
Pre- and Post-Mission Subjective Ratings	Pre-Mission Trust Survey	**Significant difference (p = 0.003)**	Indistinguishable (p = 0.321)	N/A
	Confidence	Indistinguishable (p = 0.108)	**Significant difference (p < 0.001)**	Indistinguishable (p = 0.138)
	Workload	Indistinguishable (p = 0.572)	Indistinguishable (p = 0.480)	Indistinguishable (p = 0.723)
	Satisfaction with AS plans	Indistinguishable (p = 0.585)	Indistinguishable (p = 0.556)	Indistinguishable (p = 0.554)
	Post-Experiment Trust Survey	Indistinguishable (p = 0.371)	Indistinguishable (p = 0.470)	N/A

Other Demographic Effects on Performance

A set of linear regression analyses was performed to see if there were any significant predictor variables for high (or low) system performance, operator workload, and average trust. The linear regression estimates coefficients of a linear equation, with one or more predictor variables, that best predict the value of the dependent variable. As there would be 6 linear regressions, the typical $\alpha = 0.05$ significance level was reduced to $\alpha = 0.008$ using the Bonferroni correction (Kutner et al., 2004). A backwards elimination linear regression was utilized, which removed predictor variables that did not meet a significance level of $\alpha = 0.008$, so that the most parsimonious model was derived for predicting the dependent variables. Potential predictor variables were age, gender, frequency of gaming, perception of UAVs, comfort with computers, recent amount of sleep, frequency of TV watching, education level, and Metacognitive Awareness (MAI).

The results from the 6 backwards elimination linear regressions are shown in Table 45, including the variables that were significant predictors of the performance, workload, and trust metrics. The normality, homogeneity of variance, linearity, and independence assumptions of a linear regression were met by three of the four regressions that found significant predictor variables, with the exception of Targets Found, which violates the normality assumption. Also, there was a weak to moderate correlation between two of the dependent variables: Utilization and Targets Found ($\rho=0.338$, $p = 0.001$). There were no significant predictor variables for the percentage of time targets were tracked and the number of hostiles destroyed.

Table 45. Linear Regression Results

Dependent Variable	R^2	β_0	Gaming	Military	TV	MAI
Area Coverage	0.081	$\beta = 0.875$ $p < 0.001$	-	-	-	$\beta= -0.285$ $p = 0.005$
Targets Found	0.089	$\beta = 0.925$ $p < 0.001$	-	-	$\beta= -0.298$ $p = 0.003$	-
Time Targets Tracked	0	$\beta = 0.879$ $p < 0.001$	-	-	-	-
Hostiles Destroyed	0	$\beta = 3.146$ $p < 0.001$	-	-	-	-
Utilization	0.202	$\beta = 0.523$ $p < 0.001$	$\beta= -0.450$ $p < 0.001$	-	-	-
12-Question Trust Survey	0.095	$\beta= 27.851$ $p < 0.001$	-	$\beta= -0.308$ $p = 0.002$	-	-

References

Adams, B., Bruyn, L., Houde, S., & Angelopoulos, P. (2003). *Trust in Automated Systems: Literature Review*. Defence Research and Development Canada. Report # DRDC-TORONTO-CR-2003-096.

Alighanbari, M., & How, J. P. (2006). *Robust Decentralized Task Assignment for Cooperative UAVs.* Paper presented at the AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO.

Anderson, D., Anderson, E., Lesh, N., Marks, J., Mirtich, B., Ratajczak, D., & Ryall, K. (2000). *Human-Guided Simple Search.* Paper presented at the Seventeenth Conference on Artificial Intelligence (AAAI-00), Austin, TX.

Anderson, J. R. (1983). A Spreading Activation Theory of Memory. *Journal of Verbal Learning and Verbal Behavior, 22*(3), 261-295.

Anderson, J. R. (2007). *How Can the Human Mind Occur in the Physical Universe?* New York, NY: Oxford University Press.

Ball, J., Gluck, K., Krusmark, M., Purtee, M., & Rodgers, S. (2002). *Process and Challenges in Development of the Predator Air Vehicle Operator Model.* Paper presented at the ACT-R Workshop, Pittsburgh, PA.

Barlas, Y. (1994). *Model Validation in System Dynamics.* Paper presented at the International System Dynamics Conference, Stirling, Scotland.

Beck, H. P., Dzindolet, M. T., & Pierce, L. G. (2005). *Take the Advice of a Decision Aid: I'd Rather be Wrong!* Paper presented at the Human Factors and Ergonomics Society Annual Meeting, Orlando, FL.

Bellingham, J., Richards, A., & How, J. P. (2002). *Receding Horizon Control of Autonomous Aerial Vehicles.* Paper presented at the American Controls Conference, Anchorage, AK.

Bertsimas, D., & Thiele, A. (2006). A Robust Optimization Approach to Inventory Theory. *Operations Research, 54*(1), 150-168.

Bertuccelli, L. F., Choi, H.-L., Cho, P., & How, J. P. (2009). *Real-Time Multi-UAV Task Assignment in Dynamic and Uncertain Environments.* Paper presented at the AIAA Guidance, Navigation, and Control Conference, Chicago, IL.

Bracha, S. H. (2004). Freeze, Flight, Fight, Fright, Faint: Adaptationist Perspectives on the Acute Stress Response Spectrum. *CNS Spectrums, 9*(9), 679-685.

Brehmer, B. (1990). Strategies in Real-time, Dynamic Decision Making. In R. M. Hogarth (Ed.), *Insights in Decision Making: A Tribute to Hillel J. Einhorn* (pp. 262-279). Chicago, IL: University of Chicago Press.

Bruni, S., Marquez, J. J., Brzezinski, A., Nehme, C., & Boussemart, Y. (2007). *Introducing a Human-Automation Collaboration Taxonomy (HACT) in Command and Control Decision-Support Systems.* Paper presented at the International Command and Control Research and Technology Symposium, Newport, RI.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Cave, C. B. (1997). Very Long-lasting Priming in Picture Naming. *Psychological Science, 8*(4), 322-325.

Chen, J. Y. C., Barnes, M. J., & Harper-Sciarini, M. (2011). Supervisory Control of Multiple Robots: Human-Performance Issues and User-Interface Design. *IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews, 41*(4), 435-454.

Chen, J. Y. C., Haas, E. C., & Barnes, M. J. (2007). Human Performance Issues and User Interface Design for Teleoperated Robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 37*(6), 1231-1245.

Chen, J. Y. C., & Terrence, P. I. (2009). Effects of Imperfect Automation on Concurrent Performance of Military and Robotics Tasks in a Simulated Multi-tasking Environment. *Ergonomics, 52*(8), 907-920.

Choi, H.-L., Brunet, L., & How, J. P. (2009). Consensus-Based Decentralized Auctions for Robust Task Allocation. *IEEE Transactions on Robotics, 25*(4), 912-926.

Choi, H., Kim, Y., & Kim, H. (2011). Genetic Algorithm Based Decentralized Task Assignment for Multiple Unmanned Aerial Vehicles in Dynamic Environments. *International Journal of Aeronautical & Space Science, 12*(2), 163–174.

Clare, A. S. (2010). *Dynamic Human-Computer Collaboration in Real-time Unmanned Vehicle Scheduling.* Masters of Science Thesis. Massachusetts Institute of Technology, Cambridge, MA.

Clare, A. S., & Cummings, M. L. (2011). *Task-Based Interfaces for Decentralized Multiple Unmanned Vehicle Control.* Paper presented at the AUVSI Unmanned Systems North America, Washington D.C.

Clare, A. S., Cummings, M. L., & Bertuccelli, L. F. (2012). *Identifying Suitable Algorithms for Human-Computer Collaborative Scheduling of Multiple Unmanned Vehicles.* Paper presented at the AIAA Aerospace Sciences Meeting, Nashville, TN.

Clare, A. S., Cummings, M. L., How, J., Whitten, A., & Toupet , O. (2012). Operator Objective Function Guidance for a Real-time Unmanned Vehicle Scheduling Algorithm. *AIAA Journal of Aerospace Computing, Information and Communication, 9*(4), 161-173. doi: 10.2514/1.I010019

Clare, A. S., Hart, C. S., & Cummings, M. L. (2010). *Assessing Operator Workload and Performance in Expeditionary Multiple Unmanned Vehicle Control.* Paper presented at the 48th AIAA Aerospace Sciences Meeting, Orlando, FL.

Clare, A. S., Macbeth, J. C., & Cummings, M. L. (2012). *Mixed-Initiative Strategies for Real-time Scheduling of Multiple Unmanned Vehicles.* Paper presented at the American Controls Conference, Montreal, Canada.

Clare, A. S., Maere, P. C. P., & Cummings, M. L. (2012). Assessing Operator Strategies for Real-time Replanning of Multiple Unmanned Vehicles. *Intelligent Decision Technologies, 6*(3), 221-231.

Clark, L. (2012). *18X Pilots Learn RPAs First*. Retrieved April 7, 2013, from http://www.holloman.af.mil/news/story.asp?id=123289389

Cooper, J., & Goodrich, M. A. (2008). *Towards Combining UAV and Sensor Operator Roles in UAV-Enabled Visual Search.* Paper presented at the 3rd ACM/IEEE International Conference on Human-Robot Interaction, Amsterdam, Netherlands.

Coyle, J. M., Exelby, D., & Holt, J. (1999). System Dynamics in Defence Analysis: Some Case Studies. *Journal of the Operational Research Society, 50*, 372-382.

Crandall, J. W., & Cummings, M. L. (2007). *Attention Allocation Efficiency in Human-UV Teams*. Paper presented at the AIAA Infotech@Aerospace Conference, Rohnert Park, CA.

Cummings, M., Buchin, M., Carrigan, G., & Donmez, B. (2010). Supporting Intelligent and Trustworthy Maritime Path Planning Decisions. *International Journal of Human-Computer Studies, 68*(10), 616-626.

Cummings, M. L. (2004a). *Automation Bias in Intelligent Time Critical Decision Support Systems.* Paper presented at the AIAA 3rd Intelligent Systems Conference, Chicago, IL.

Cummings, M. L. (2004b). The Need for Command and Control Instant Message Adaptive Interfaces: Lessons Learned from Tactical Tomahawk Human-in-the-Loop Simulations. *Cyberpsychology and Behavior, 7*(6), 653-661.

Cummings, M. L., & Bruni, S. (2009). Collaborative Human-Automation Decision Making. In S. Y. Nof (Ed.), *Handbook of Automation LXXVI* (pp. 437-448). New York, NY: Springer.

Cummings, M. L., Bruni, S., & Mitchell, P. J. (2010). Human Supervisory Control Challenges in Network-Centric Operations. *Reviews of Human Factors and Ergonomics, 6*(1), 34-78.

Cummings, M. L., Clare, A. S., & Hart, C. S. (2010). The Role of Human-Automation Consensus in Multiple Unmanned Vehicle Scheduling. *Human Factors, 52*(1), 17-27.

Cummings, M. L., & Guerlain, S. (2004). Using a Chat Interface as an Embedded Secondary Tasking Tool. In D. A. Vincenzi, M. Mouloua & P. A. Hancock (Eds.), *Human Performance, Situation Awareness and Automation: Current Research and Trends. HPSAA II* (Vol. 1, pp. 240-248). London: Routledge.

Cummings, M. L., & Guerlain, S. (2007). Developing Operator Capacity Estimates for Supervisory Control of Autonomous Vehicles. *Human Factors, 49*(1), 1-15.

Cummings, M. L., How, J., Whitten, A., & Toupet, O. (2012). The Impact of Human-Automation Collaboration in Decentralized Multiple Unmanned Vehicle Control. *Proceedings of the IEEE, 100*(3), 660-671.

Cummings, M. L., Mastracchio, C., Thornburg, K. M., & Mkrtchyan, A. (2013). Boredom and Distraction in Multiple Unmanned Vehicle Supervisory Control. *Interacting with Computers, 25*(1), 34-47. doi: 10.1093/iwc/iws011

Cummings, M. L., & Mitchell, P. J. (2008). Predicting Controller Capacity in Supervisory Control of Multiple UAVs. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 38*(2), 451-460.

Cummings, M. L., & Nehme, C. E. (2010). Modeling the Impact of Workload in Network Centric Supervisory Control Settings. In S. Kornguth, R. Steinberg & M. D. Matthews (Eds.), *Neurocognitive and Physiological Factors During High-tempo Operations* (pp. 23-40). Surrey, UK: Ashgate.

Cummings, M. L., Nehme, C. E., & Crandall, J. (2007). Predicting Operator Capacity for Supervisory Control of Multiple UAVs. In J. Chahl, L. Jain, A. Mizutani & M. Sato-Ilic (Eds.), *Innovations in Intelligent Machines - 1* (Vol. 70, pp. 11-37). Berlin: Springer.

Cummings, M. L., & Thornburg, K. T. (2011). Paying Attention to the Man Behind the Curtain. *IEEE Pervasive Computing, 10*(1), 58-62.

de Visser, E., Jacobs, R., Chabuk, T., Freedy, A., & Scerri, P. (2012). *Design and Evaluation of the Adaptive Interface Management System (AIMS) for Collaborative Mission Planning with Unmanned Vehicles.* Paper presented at the AIAA Infotech@Aerospace 2012, Garden Grove, CA.

de Visser, E. J., Krueger, F., McKnight, P., Scheid, S., Smith, M., Chalk, S., & Parasuraman, R. (2012). *The World is not Enough: Trust in Cognitive Agents.* Paper presented at the Human Factors and Ergonomics Society Annual Meeting, Boston, MA.

Dimperio, E., Gunzelmann, G., & Harris, J. (2008). *An Initial Evaluation of a Cognitive Model of UAV Reconnaissance.* Paper presented at the Seventeenth Conference on Behavior Representation in Modeling and Simulation, Orlando, FL.

Dixon, S. R., & Wickens, C. D. (2003). *Control of Multiple-UAVs: A Workload Analysis.* Paper presented at the 12th International Symposium on Aviation Psychology, Dayton, OH.

DoD. (2011). *Unmanned Aircraft Systems Integrated Roadmap 2011-2036.* Office of the Secretary of Defense.

Domke, D., Shah, D. V., & Wackman, D. B. (1998). Media Priming Effects: Accessibility, Association, and Activation. *International Journal of Public Opinion Research, 10*(1), 51-74.

Donmez, B. D., Nehme, C., & Cummings, M. L. (2010). Modeling Workload Impact in Multiple Unmanned Vehicle Supervisory Control. *IEEE Systems, Man, and Cybernetics, Part A Systems and Humans, 40*(6), 1180-1190.

Drury, J. L., Scholtz, J., & Kieras, D. (2007). *Adapting GOMS to Model Human-Robot Interaction.* Paper presented at the ACM/IEEE International Conference on Human-Robot Interaction, Washington, D.C.

Dzindolet, M. T., Pierce, L. G., Beck, H. P., & Dawe, L. A. (2002). The Perceived Utility of Human and Automated Aids in a Visual Detection Task. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 44*(1), 79-94.

Endsley, M. R. (1995). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors, 37*(1), 32-64.

Endsley, M. R., Sollenberger, R., & Stein, E. (2000). *Situation Awareness: A Comparison of Measures.* Paper presented at the Human Performance, Situation Awareness and Automation: User Centered Design for the New Millennium Conference, Savannah, GA.

Eun, Y., & Bang, H. (2007). *Cooperative Task Assignment and Path Planning of Multiple UAVs Using Genetic Algorithm.* Paper presented at the AIAA Infotech@Aerospace 2007 Conference, Rohnert Park, CA.

Fisher, S. (2008). *Replan Understanding for Heterogenous Unmanned Vehicle Teams.* Masters of Engineering Thesis. Massachusetts Institute of Technology, Cambridge, MA.

Forrester, J. W. (1961). *Industrial Dynamics.* Cambridge, MA: MIT Press.

Forrester, J. W., & Senge, P. M. (1980). Tests for Building Confidence in System Dynamics Models. *TIMS Studies in the Management Sciences, 14*(1), 209-228.

Gao, F., Clare, A. S., Macbeth, J. C., & Cummings, M. L. (2013). *Modeling the Impact of Operator Trust on Performance in Multiple Robot Control.* Paper presented at the AAAI Spring Symposium: Trust and Autonomous Systems, Stanford, CA.

Gao, F., Cummings, M. L., & Bertuccelli, L. F. (2012). *Teamwork in Controlling Multiple Robots.* Paper presented at the ACM/IEEE International Conference on Human-Robot Interaction (HRI '12), Boston, MA.

Gao, J., & Lee, J. D. (2006). Extending the Decision Field Theory to Model Operators' Reliance on Automation in Supervisory Control Situations. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans, 36*(5), 943-959.

Geramifard, A., Redding, J. D., Joseph, J., Roy, N., & How, J. P. (2012). *Model Estimation Within Planning and Learning.* Paper presented at the American Control Conference, Montreal, Canada.

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception.* Boston, MA: Houghton Mifflin.

Girard, A. R., & Hedrick, J. K. (2004). *Border Patrol and Surveillance Missions Using Multiple Unmanned Air Vehicles.* Paper presented at the 43rd IEEE Conference on Decision and Control, Paradise Islands, The Bahamas.

Goodrich, M. A., Johansen, J., McLain, T. W., Anderson, J., Sun, J., & Crandall, J. W. (2007). *Managing Autonomy in Robot Teams: Observations from Four Experiments.* Paper presented at the 2nd ACM/IEEE International Conference on Human Robot Interaction, Washington D.C.

Goodrich, M. A., Olsen, D. R., Crandall, J. W., & Palmer, T. J. (2001). *Experiments in Adjustable Autonomy.* Paper presented at the International Joint Conference on Artificial Intelligence - Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents, Seattle, WA.

Goodrich, M. A., Quigley, M., & Cosenzo, K. (2005). *Task Switching and Multi-Robot Teams.* Paper presented at the 3rd International Workshop on Multi-Robot Systems, Washington D.C.

Green, C. S., & Bavelier, D. (2003). Action Video Game Modifies Visual Selective Attention. *Nature, 423*(6939), 534-537.

Größler, A., & Milling, P. (2007). *Inductive and Deductive System Dynamics Modeling.* Paper presented at the 25th International Conference of the System Dynamics Society, Boston, MA.

Guerlain, S. A. (1995). *Using the Critiquing Approach to Cope With Brittle Expert Systems.* Paper presented at the Human Factors and Ergonomics Society 39th Annual Meeting, San Diego, CA.

FAA Modernization and Reform Act of 2012, H.R. 658, 112th Cong., 2nd Sess. (2012).

Haddal, C. C., & Gertler, J. (2010). *Homeland Security: Unmanned Aerial Vehicles and Border Surveillance*. Congressional Research Service.

Hancock, P. A., Billings, D. R., Schaefer, K. E., Chen, J. Y., De Visser, E. J., & Parasuraman, R. (2011). A Meta-Analysis of Factors Affecting Trust in Human-Robot Interaction. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 53*(5), 517-527.

Hart, S., & Staveland, L. (1988). Development of the NASA-TLX: Results of Empirical and Theoretical Research. In P. A. Hancock & N. Meshkati (Eds.), *Human Mental Workload* (pp. 139-183). Amsterdam: North Holland.

Henson, R. N. A. (2003). Neuroimaging Studies of Priming. *Progress in Neurobiology, 70*(1), 53-81.

Hoffman, R. R., Johnson, M., Bradshaw, J. M., & Underbrink, A. (2013). Trust in Automation. *IEEE Intelligent Systems, 28*(1), 84-88.

Homer, J. B. (1983). *Partial-Model Testing as a Validation Tool for System Dynamics.* Paper presented at the 1983 International System Dynamics Conference, Chestnut Hill, MA.

How, J. P., Fraser, C., Kulling, K. C., Bertuccelli, L. F., Toupet, O., Brunet, L., Bachrach, A., & Roy, N. (2009). Increasing Autonomy of UAVs. *IEEE Robotics & Automation Magazine, 16*(2), 43-51.

Howe, A. E., Whitley, L. D., Barbulescu, L., & Watson, J. P. (2000). *Mixed Initiative Scheduling for the Air Force Satellite Control Network.* Paper presented at the 2nd International NASA Workshop on Planning and Scheduling for Space, San Francisco, CA.

Jackson, S., Wilson, J. R., & MacCarthy, B. L. (2004). A New Model of Scheduling in Manufacturing: Tasks, Roles, and Monitoring. *Human Factors, 46*(3), 533-550.

Jenkins, D. (2013). *The Economic Impact of Unmanned Aircraft Systems Integration in the United States*. Association for Unmanned Vehicle Systems International (AUVSI) Economic Report.

Jian, J., Bisantz, A. M., & Drury, C. G. (2000). Foundations for an Empirically Determined Scale of Trust in Automated Systems. *International Journal of Cognitive Ergonomics, 4*(1), 53-71.

Johnson, K., Ren, L., Kuchar, J. K., & Oman, C. M. (2002). *Interaction of Automation and Time Pressure in a Route Replanning Task.* Paper presented at the International Conference on Human-Computer Interaction in Aeronautics (HCI-Aero), Cambridge, MA.

Kahneman, D. (1973). *Attention and Effort.* Englewood Cliffs, NJ: Prentice-Hall.

Kant, R. (2012). A Study of Creativity of Secondary School Children as a Correlate of Some Television Viewing Habits. *International Journal of Modern Education and Computer Science, 4*(10), 33-39.

Khasawneh, M. T., Bowling, S. R., Jiang, X., Gramopadhye, A. K., & Melloy, B. J. (2003). *A Model for Predicting Human Trust in Automated Systems.* Paper presented at the International Conference on Industrial Engineering – Theory, Applications and Practice, Las Vegas, NV.

Klau, G. W., Lesh, N., Marks, J., & Mitzenmacher, M. (2003). *Human-Guided Search: Survey and Recent Results*. Mitsubishi Electric Research Laboratories. Report # TR2003-07.

Kopeikin, A., Clare, A., Toupet, O., How, J. P., & Cummings, M. L. (2012). *Flight Testing a Heterogeneous Multi-UAV System with Human Supervision.* Paper presented at the AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN.

Kosslyn, S. M., & Rosenberg, R. S. (2011). *Introducing Psychology: Brain, Person, Group*. Boston: Pearson Learning Solutions.

Kutner, M., Nachtsheim, C., Neter, J., & Li, W. (2004). *Applied Linear Statistical Models* (5th ed.). Boston: McGraw-Hill Irwin.

Kwon, O. K., Martland, C. D., & Sussman, J. M. (1998). Routing and Scheduling Temporal and Heterogeneous Freight Car Traffic on Rail Networks. *Transportation Research Part E: Logistics and Transportation Review, 34*(2), 101-115

Lacson, F. C., Wiegmann, D. A., & Madhavan, P. (2005). *Effects of Attribute and Goal Framing on Automation Reliance and Compliance*. Paper presented at the Human Factors and Ergonomics Society Annual Meeting, New Orleans, LA.

Layton, C., Smith, P. J., & McCoy, C. E. (1994). Design of a Cooperative Problem-solving System for En-route Flight Planning - An Empirical Evaluation. *Human Factors, 36*(1), 94-116. doi: 10.1177/001872089403600106

Lee, J. D., & Moray, N. (1992). Trust, Control Strategies and Allocation of Functions in Human-Machine Systems. *Ergonomics, 35*(10), 1234-1270.

Lee, J. D., & Moray, N. (1994). Trust, Self-Confidence, and Operators' Adaptation to Automation. *International Journal of Human-Computer Studies, 40*(1), 153-184.

Lee, J. D., & See, K. A. (2004). Trust in Automation: Designing for Appropriate Reliance. *Human Factors, 46*(1), 50-80. doi: 10.1518/hfes.46.1.50_30392

Lewandowsky, S., Mundy, M., & Tan, G. P. (2000). The Dynamics of Trust: Comparing Humans to Automation. *Journal of Experimental Psychology: Applied, 6*(2), 104-123.

Lewis, M., Wang, J., & Hughes, S. (2007). USARsim: Simulation for the Study of Human-Robot Interaction. *Journal of Cognitive Engineering and Decision Making, 1*(1), 98-120.

Linegang, M. P., Stoner, H. A., Patterson, M. J., Seppelt, B. D., Hoffman, J. D., Crittendon, Z. B., & Lee, J. D. (2006). *Human-Automation Collaboration in Dynamic Mission Planning: A Challenge Requiring an Ecological Approach.* Paper presented at the 50th Annual Meeting of the Human Factors and Ergonomics Society, San Francisco, CA.

Lorch, R. F. (1982). Priming and Search Processes in Semantic Memory: A Test of Three Models of Spreading Activation. *Journal of Verbal Learning and Verbal Behavior, 21*(4), 468-492.

Madrigal, A. C. (2011). *Inside the Drone Missions to Fukushima*. Retrieved April 7, 2013, from http://www.theatlantic.com/technology/archive/2011/04/inside-the-drone-missions-to-fukushima/237981/

Malasky, J., Forest, L. M., Khan, A. C., & Key, J. R. (2005). *Experimental Evaluation of Human-Machine Collaborative Algorithms in Planning for Multiple UAVs.* Paper presented at the IEEE International Conference on Systems, Man and Cybernetics, Waikoloa, HI.

Mekdeci, B., & Cummings, M. L. (2009). *Modeling Multiple Human Operators in the Supervisory Control of Heterogeneous Unmanned Vehicles.* Paper presented at the Conference on Performance Metrics for Intelligent Systems, Gaithersburg, MD.

Michini, B., Cutler, M., & How, J. P. (2013). *Scalable Reward Learning from Demonstration*. Paper presented at the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany.

Miller, C. (2004). Human-Computer Etiquette: Managing Expectations with Intentional Agents. *Communications of the ACM, 47*(4), 31-34.

Miller, C., Funk, H., Wu, P., Goldman, R., Meisner, J., & Chapman, M. (2005). *The Playbook Approach to Adaptive Automation.* Paper presented at the Human Factors and Ergonomics Society 49th Annual Meeting, Orlando, FL.

Miller, C. A., & Parasuraman, R. (2003). *Beyond Levels of Automation: An Architecture for More Flexible Human-Automation Collaboration.* Paper presented at the Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Denver, CO.

Miller, J. E., & Perkins, L. (2010). *Development of Metrics for Trust in Automation.* Paper presented at the International Command and Control Research and Technology Symposium (ICCRTS), Santa Monica, CA.

Miller, J. G. (1978). Information Input Overload. *Living Systems* (pp. 121-202). New York: McGraw-Hill.

Miyata, Y., & Norman, D. A. (1986). Psychological Issues in Support of Multiple Activities. In D. A. Norman & S. W. Draper (Eds.), *User Centered System Design: New Perspectives on Human Computer Interaction* (pp. 265-284). Hillsdale, NJ: Lawrence Erlbaum.

Mkrtchyan, A. A. (2011). *Modeling Operator Performance in Low Task Load Supervisory Domains.* Masters of Science Thesis. Massachusetts Institute of Technology, Cambridge, MA.

Mohan, Y., & Ponnambalam, S. (2009). *An Extensive Review of Research in Swarm Robotics*. Paper presented at the World Congress on Nature & Biologically Inspired Computing, Coimbatore, India.

Moray, N., Inagaki, T., & Itoh, M. (2000). Adaptive Automation, Trust, and Self-Confidence in Fault Management of Time-Critical Tasks. *Journal of Experimental Psychology: Applied, 6*(1), 44-58.

Mosier, K., Skitka, L. J., Heers, S., & Burdick, M. D. (1998). Automation Bias: Decision Making and Performance in High-tech Cockpits. *International Journal of Aviation Psychology, 8*(1), 47-63.

Muir, B. M. (1987). Trust Between Humans and Machines, and the Design of Decision Aids. *International Journal of Man-Machine Studies, 27*(5&6), 527-539.

Muir, B. M., & Moray, N. (1996). Trust in Automation. Part II. Experimental Studies of Trust and Human Intervention in a Process Control Simulation. *Ergonomics, 39*(3), 429-460.

Naill, R. (1973). The Discovery Life Cycle of a Finite Resource: A Case Study of U.S. Natural Gas. In D. L. Meadows & D. H. Meadows (Eds.), *Toward Global Equilibrium: Collected Papers* (pp. 213-256). Waltham, MA: Pegasus Communications.

NASA. (2012). *NASA Global Hawks Aid UAV-to-UAV Refueling Project*. Retrieved April 7, 2013, from http://www.nasa.gov/centers/dryden/status_reports/global_hawk_status_10_05_12.html

Naval Studies Board. (2005). *Autonomous Vehicles in Support of Naval Operations*. Washington D.C.: National Research Council.

Nehme, C. (2009). *Modeling Human Supervisory Control in Heterogeneous Unmanned Vehicle Systems.* Ph.D. Thesis. Massachusetts Institute of Technology, Cambridge, MA.

Niedeggen, M., & Rösler, F. (1999). N400 Effects Reflect Activation Spread During Retrieval of Arithmetic Facts. *Psychological Science, 10*(3), 271-276.

Office of the Secretary of Defense. (2005). *Unmanned Aircraft Systems (UAS) Roadmap, 2005-2030*. Washington D.C.: DoD.

Olsen, D. R., & Wood, S. B. (2004). *Fan-out: Measuring Human Control of Multiple Robots*. Paper presented at the SIGCHI conference on Human factors in Computing Systems, Vienna, Austria.

Özgün, O., & Barlas, Y. (2009). *Discrete vs. Continuous Simulation: When Does It Matter?* Paper presented at the 27th International Conference of The System Dynamics Society, Albuquerque, NM.

Parasuraman, R. (1993). Effects of Adaptive Function Allocation on Human Performance. In D. J. Garland & J. A. Wise (Eds.), *Human Factors and Advanced Aviation Technologies* (pp. 147-157). Daytona Beach, FL: Embry-Riddle Aeronautical University Press.

Parasuraman, R. (2000). Designing Automation for Human Use: Empirical Studies and Quantitative Models. *Ergonomics, 43*(7), 931-951.

Parasuraman, R., & Riley, V. (1997). Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors, 39*(2), 230-253.

Polson, P., & Smith, N. (1999). *The Cockpit Cognitive Walkthrough.* Paper presented at the 10th Symposium on Aviation Psychology, Columbus, OH.

Ponda, S., Ahmed, N., Luders, B., Sample, E., Hoossainy, T., Shah, D., Campbell, M., & How, J. (2011). *Decentralized Information-Rich Planning and Hybrid Sensor Fusion for Uncertainty Reduction in Human-Robot Missions.* Paper presented at the AIAA Guidance, Navigation, and Control Conference, Portland, OR.

Powell, M. J. D. (1964). An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives. *The Computer Journal, 7*(2), 155-162. doi: 10.1093/comjnl/7.2.155

Rad, K. N. (2008). *Modelling Collaborative Planning and Scheduling Scenarios that Include Human and Computer Decision Activities.* Ph.D. Thesis. Swinburne University of Technology, Melbourne, Australia.

Rasmussen, J. (1976). Outlines of a Hybrid Model of the Process Plant Operator. In T. B. Sheridan & G. Johannsen (Eds.), *Monitoring Behavior and Supervisory Control*. New York, NY: Plenum Press.

Rasmussen, J. (1983). Skills, Rules, and Knowledge: Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models. *IEEE Transactions on Systems, Man, and Cybernetics, 13*(3), 257-266.

Ricard, G. L. (1994). Manual Control With Delays: A Bibliography. *ACM SIGGRAPH Computer Graphics, 28*(2), 149-154.

Rice, S., Clayton, K., Wells, A., & Keller, D. (2008). *Manipulating Trust Behaviors in a Combat Identification Task.* Paper presented at the Combat Identification Workshop, Gold Canyon, AZ.

Richardson, G. P., & Pugh, A. L. (1981). *Introduction to System Dynamics Modeling with DYNAMO*. Cambridge, MA: Productivity Press.

Riley, V. (1989). *A General Model of Mixed-Initiative Human-Machine Systems.* Paper presented at the Human Factors and Ergonomics Society 33rd Annual Meeting, Denver, CO.

Rodas, M. O., Veronda, M., & Szatkowski, C. (2011). *Developing a Decision Tool to Evaluate Unmanned System's Command and Control Technologies in Network Centric Operations Environments.* Paper presented at the The Third International Conference on Advanced Cognitive Technologies and Applications, Rome, Italy.

Rouse, W. B. (1983). *Systems Engineering Models of Human-Machine Interaction*. New York: Elsevier North Holland.

Rubio, S., Diaz, E., Martin, J., & Puente, J. M. (2004). Evaluation of Subjective Mental Workload: A Comparison of SWAT, NASA-TLX, and Workload Profile Methods. *Applied Psychology: An International Review, 53*(1), 61-86.

Rudolph, J. W., Morrison, J. B., & Carroll, J. S. (2009). The Dynamics of Action-Oriented Problem Solving: Linking Interpretation and Choice. *Academy of Management Review, 34*(4), 733-756.

Rudolph, J. W., & Repenning, N. P. (2002). Disaster Dynamics: Understanding the Role of Quantity in Organizational Collapse. *Administrative Science Quarterly, 47*(1), 1-30. doi: 10.2307/3094889

Ruff, H. A., Narayanan, S., & Draper, M. H. (2002). Human Interaction with Levels of Automation and Decision-Aid Fidelity in the Supervisory Control of Multiple Simulated Unmanned Air Vehicles. *Presence: Teleoperators and Virtual Environments, 11*(4), 335-351. doi: 10.1162/105474602760204264

Russell, S., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall.

Ryan, J. C. (2011). *Assessing the Performance of Human-Automation Collaborative Planning Systems.* Masters of Science Thesis. Massachusetts Institute of Technology, Cambridge, MA.

Ryan, J. C., Banerjee, A. G., Cummings, M. L., & Roy, N. (2011). Comparing the Performance of Expert User Heuristics and an Integer Linear Program in Aircraft Carrier Deck Operations. *Journal of Heuristics, 17*(5), 1-25.

Sanders, T., Oleson, K. E., Billings, D. R., Chen, J. Y. C., & Hancock, P. A. (2011). *A Model of Human-Robot Trust: Theoretical Model Development.* Paper presented at the Proceedings of the Human Factors and Ergonomics Society 55th Annual Meeting, Las Vegas, NV.

Sanderson, P. M. (1991). Towards the Model Human Scheduler. *International Journal of Human Factors in Manufacturing, 1*(3), 195-219.

Sargent, R. G. (2005). *Verification and Validation of Simulation Models.* Paper presented at the 2005 Winter Simulation Conference, Orlando, Florida.

Sastry, M. A. (1995). *Time and Tide in Organizations: Simulating Change Processes in Adaptive, Punctuated, and Ecological Theories of Organizational Evolution.* Ph.D. Thesis. Massachusetts Institute of Technology, Cambridge, MA.

Savla, K., Nehme, C., Temple, T., & Frazzoli, E. (2008). *On Efficient Cooperative Strategies between UAVs and Humans in a Dynamic Environment.* Paper presented at the AIAA Guidance, Navigation and Control Conference, Honolulu, HI.

Scerbo, M. W. (2001). Stress, Workload, and Boredom in Vigilance: A Problem and an Answer. In P. A. Hancock & P. A. Desmond (Eds.), *Stress, Workload and Fatigue*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Schacter, D. L. (1987). Implicit Memory: History and Current Status. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 13*(3), 501-518.

Schacter, D. L., & Buckner, R. L. (1998). Priming and the Brain. *Neuron, 20*(2), 185-195.

Schmidt, D. K. (1978). A Queuing Analysis of the Air Traffic Controller's Workload. *IEEE Transactions on Systems, Man, and Cybernetics, 8*(6), 492-498. doi: 10.1109/TSMC.1978.4310003

Schmidt, M. E., & Vandewater, E. A. (2008). Media and Attention, Cognition, and School Achievement. *The Future of Children, 18*(1), 63-85.

Schraw, G., & Dennison, R. S. (1994). Assessing Metacognitive Awareness. *Contemporary Educational Pscyhology, 19*(4), 460-475.

Scott, S. D., Lesh, N., & Klau, G. W. (2002). *Investigating Human-Computer Optimization.* Paper presented at the Conference on Human Factors in Computer Systems (CHI), Minneapolis, MN.

See, K. A. (2002). *The Effect of Sound on Performance, Reliance, Decision Making, and Trust in Semi-Automatic Process Control Systems.* Masters of Science Thesis. University of Iowa, Iowa City, IA.

Senge, P. M. (1980). A System Dynamics Approach to Investment-Function Formulation and Testing. *Socio-Economic Planning Sciences, 14*(6), 269-280.

Shejwal, B., & Purayidathil, J. (2006). Television Viewing of Higher Secondary Students: Does It Affect Their Academic Achievement and Mathematical Reasoning? *Psychology & Developing Societies, 18*(2), 201-213.

Sheridan, T. B. (1992). *Telerobotics, Automation and Human Supervisory Control*. Cambridge, MA: The MIT Press.

Sheridan, T. B. (1993). Space Teleoperation Through Time Delay: Review and Prognosis. *IEEE Transactions on Robotics and Automation, 9*(5), 592-606.

Sheridan, T. B. (2006). Supervisory Control. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (3rd ed.). Hoboken, NJ: John Wiley & Sons, Inc.

Silverman, B. G. (1992). Human-Computer Collaboration. *Human-Computer Interaction, 7*(2), 165-196.

Simon, H. A., Hogarth, R., Piott, C. R., Raiffa, H., Schelling, K. A., Thaier, R., Tversky, A., & Winter, S. (1986). *Decision Making and Problem Solving.* Paper presented at the Research Briefings 1986: Report of the Research Briefing Panel on Decision Making and Problem Solving, Washington D.C.

Smith, P., McCoy, E., & Layton, C. (1997). Brittleness in the Design of Cooperative Problem-solving Systems: The Effects on User Performance. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 27*(3), 360-371.

Southern, D. N. (2010). *Human-Guided Management of Collaborating Unmanned Vehicles in Degraded Communication Environments.* Masters of Engineering Thesis. Massachusetts Institute of Technology, Cambridge, MA.

Squire, P., Trafton, G., & Parasuraman, R. (2006). *Human Control of Multiple Unmanned Vehicles: Effects of Interface Type on Execution and Task Switching Times.* Paper presented at the ACM Conference on Human-Robot Interaction, Salt Lake City, UT.

Stankovic, J. A. (1988). Misconceptions About Real-Time Computing: A Serious Problem for Next-Generation Systems. *Computer, 21*(10), 10-19.

Sterman, J. D. (1984). Appropriate Summary Statistics for Evaluating the Historical Fit of System Dynamics Models. *Dynamica, 10*(Winter), 51-86.

Sterman, J. D. (1987a). Expectation Formation in Behavioral Simulation Models. *Behavioral Science, 32*(3), 190-211.

Sterman, J. D. (1987b). Testing Behavioral Simulation Models by Direct Experiment. *Management Science, 33*(12), 1572-1592.

Sterman, J. D. (1989a). Misperceptions of Feedback in Dynamic Decision Making. *Organizational Behavior and Human Decision Processes, 43*(3), 301-335.

Sterman, J. D. (1989b). Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment. *Management Science, 35*(3), 321-339.

Sterman, J. D. (2000). *Business Dynamics: Systems Thinking and Modeling for a Complex World*. New York: Irwin/McGraw-Hill.

Sujit, P. B., George, J. M., & Beard, R. (2008). *Multiple UAV Task Allocation Using Particle Swarm Optimization.* Paper presented at the AIAA Guidance, Navigation, and Control Conference and Exhibit, Honolulu, HI.

Sweetser, A. (1999). *A Comparison of System Dynamics (SD) and Discrete Event Simulation (DES).* Paper presented at the 17th International Conference of the System Dynamics Society and 5th Australian & New Zealand Systems Conference, Wellington, New Zealand.

Theil, H. (1966). *Applied Economic Forecasting*. Amsterdam: North Holland Publishing Company.

Thomas, R. (2012). *Humanitarian Support and Disaster Relief*. Paper presented at the UAS Action Summit, Grand Forks, ND.

Thompson, J. M., Ottensmeyer, M. P., & Sheridan, T. B. (1999). Human Factors in Telesurgery: Effects of Time Delay and Asynchrony in Video and Control Feedback With Local Manipulative Assistance. *Telemedicine Journal, 5*(2), 129-137.

Thorner, J. L. (2007). *Trust-Based Design of Human-Guided Algorithms.* Master of Science Thesis. Massachusetts Institute of Technology, Cambridge, MA.

Tulga, M. K., & Sheridan, T. B. (1980). Dynamic Decisions and Work Load in Multitask Supervisory Control. *IEEE Transactions on Systems, Man, and Cybernetics, 10*(5), 217-232.

Tushman, M., & Romanelli, E. (1985). Organizational Evolution: A Metamorphosis Model of Convergence and Reorientation. In B. M. Staw & L. L. Cummings (Eds.), *Research in Organizational Behavior* (Vol. 7, pp. 171-222). Greenwich, CT: JAI Press.

Tversky, A., & Kahneman, D. (1974). Judgment Under Uncertainty: Heuristics and Biases. *Science, 185*(4157), 1124-1131.

U.S. Air Force. (2009). *United States Air Force Unmanned Aircraft Systems Flight Plan 2009-2047*. Washington, D.C.: United States Air Force.

U.S. Department of Defense. (1999). *Department of Defense Interface Standard: Common Warfighting Symbology*. Report # MIL-STD-2525B.

U.S. Department of Defense. (2012). *The Role of Autonomy in DoD Systems*. Defense Science Board Task Force Report.

Walker, P., Nunnally, S., Lewis, M., Kolling, A., Chakraborty, N., & Sycara, K. (2012). *Neglect Benevolence in Human Control of Swarms in the Presence of Latency*. Paper presented at the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, Korea.

Walters, B., French, J., & Barnes, M. J. (2000). *Modeling the Effect of Crew Size and Crew Fatigue on the Control of Tactical Unmanned Aerial Vehicles (TUAVs)*. Paper presented at the Winter Simulation Conference, Orlando, FL.

Washington Technology. (2009). *DOD Requests $5.4B for Unmanned Systems Budget*. Retrieved June 16, 2011, from http://washingtontechnology.com/articles/2009/06/16/unmanned-system-spending.aspx

White, A. S. (2003). *A Qualitative Systems Dynamic Model of Human Monitoring of Automated Systems.* Paper presented at the 21st International Conference of the Systems Dynamics Society, New York, NY.

Whitten, A. K. (2010). *Decentralized Planning for Autonomous Agents Cooperating in Complex Missions.* Masters of Science Thesis. Massachusetts Institute of Technology, Cambridge, MA.

Wickens, C. D., & Hollands, J. G. (2000). *Engineering Psychology and Human Performance* (3rd ed.). Upper Saddle River, N.J.: Prentice Hall.

Wickens, C. D., Mavor, A., Parasuraman, R., & McGee, J. M. (1998). *The Future of Air Traffic Control: Human Operators and Automation*. Washington D.C.: National Academy Press.

Wilms, I. L., Petersen, A., & Vangkilde, S. (2013). Intensive Video Gaming Improves Encoding Speed to Visual Short-Term Memory in Young Male Adults. *Acta Psychologica, 142*(1), 108-118.

Wright, T. P. (1936). Factors Affecting the Cost of Airplanes. *Journal of Aeronautical Sciences, 3*(4), 122-128.

Yerkes, R. M., & Dodson, J. D. (1908). The Relation of Strength of Stimulus to Rapidity of Habit-Formation. *Journal of Comparative Neurology and Psychology, 18*, 459-482.